Oracle® Fusion Middleware

Administering Oracle Privileged Account Manager

11*g* Release 2 (11.1.2.3)

E52312-02

April 2016

Documentation for administrators and end users that describes how to use Oracle Privileged Account Manager to administer, audit, and provide better security for privileged accounts and passwords in your organization.

ORACLE

Oracle Fusion Middleware Administering Oracle Privileged Account Manager, 11*g* Release 2 (11.1.2.3)

E52312-02

Primary Author: Gauhar Khan

Contributing Authors: K. C. Francis, Prakash Hulikere, Gowri G.R

Contributors: Arun Theebaprakasam, Kwan-I Lee, Ayush Jindal, Fannie Ho, Himanshu S. Sharma, Olaf Stullich, Daniel Shih, An Li, Vishal Mishra, Mark Wilcox, Stella Mao, Zhe Wang, Sudhir Kumar Srinivasan, and Iris Li

Contents

6 Working with Targets

7 Working with Service Accounts

8 Configuring and Managing Agents

9 Working with Privileged Accounts

13 Working with Plug-Ins

14 Working with Self-Service

Part III Monitoring Oracle Privileged Account Manager

15 Working with Reports

16 Managing Oracle Privileged Account Manager Auditing and Logging

19 Integrating Oracle Privileged Account Manager with Other Oracle Identity Management Components

20 Troubleshooting Oracle Privileged Account Manager

B Working with Oracle Privileged Account Manager's RESTful Interface

List of Examples

List of Figures

List of Tables

Preface

Welcome to *Administering Oracle Privileged Account Manager*. This guide describes how to use and administer Oracle Privileged Account Manager in an enterprise infrastructure.

Audience

The *Administering Oracle Privileged Account Manager* is intended for Oracle Privileged Account Manager administrators who can configure connections to target systems and client applications, access passwords for target systems, and who can create roles and assign users to those roles.

Administrators must be familiar with either the UNIX operating system or the Microsoft Windows operating system to understand the command-line syntax and examples in this document. You also must be familiar with the Lightweight Directory Access Protocol (LDAP).

This guide is also intended for Oracle Privileged Account Manager end-users who do not have administrative privileges, but who are authorized to check privileged accounts in and out.

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 in the Oracle Identity Manager 11*g* Release 2 (11.1.2.3) documentation set:

- *Oracle Database Advanced Security Administrator's Guide*
- *Oracle Database Vault Administrator's Guide*
- *Oracle Fusion Middleware Administrator's Guide*

- *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Manager*
- *Oracle Fusion Middleware Application Security Guide*
- *Oracle Fusion Middleware Command Reference for Oracle WebLogic Server*
- *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*
- *Oracle Fusion Middleware Developing Applications for Oracle WebLogic Server*
- *Oracle Fusion Middleware Error Messages Reference*
- *Oracle Fusion Middleware Installation Planning Guide*
- *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*
- *Oracle Fusion Middleware Integration Guide for Oracle Identity Management Suite*
- *Oracle Fusion Middleware Integration Overview for Oracle Identity Management Suite*
- *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*
- *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*
- *Oracle Fusion Middleware Release Notes for Oracle Identity Management*
- *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*
- *Oracle Fusion Middleware Securing Oracle WebLogic Server*
- *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management*
- *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*
- *Oracle Fusion Middleware User's Guide for Oracle Identity Manager*
- *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*
- *Oracle Identity Manager Connector Guide for Database User Management*
- *Oracle Identity Manager Connector Guide for Oracle Internet Directory*

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 product changes for Oracle Privileged Account Manager in the Oracle Fusion Middleware 11*g* Release 2 (11.1.2.3) release.

This chapter includes the following topics:

- New and Changed Features for 11g Release 2 (11.1.2.3.0)
- New and Changed Features for 11g Release 2 (11.1.2.2.0)
- Other Significant Changes in This Guide

Follow the pointers in this guide to get more information about the features and how to use them.

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

Oracle Privileged Account Manager 11g Release 2 (11.1.2.3.0) includes the following new and changed administrative, development, and security features:

- Added support for the Connector Server. Refer to Section 5.4, "Managing a Connector Server" for more information about configuring connector servers.
- Added support for the Windows target type. Refer to Chapter 6, "Working with Targets" for more information.
- Added support for Windows Session Recording using the Windows Agent. Refer to Chapter 8, "Configuring and Managing Agents" for more information about the actions a user preforms on a Windows target.
- Enhanced the Session Recording feature to include DVR-Like session recording and playback. Refer to Section 9.7, "Viewing a Session Recording" for more information about viewing session recordings.
- Added the SAP UM and SAP UME target types, which are used to manage privileged accounts on SAP software systems. Refer to the following:
  - Chapter 6, "Working with Targets" for more information about configuring SAP targets.
  - Chapter 9, "Working with Privileged Accounts" for more information about Oracle Privileged Account Manager operations (such as password checkout, password reset, and automatic password recycling) on SAP targets and accounts.
- Added support for Session and Password management for the SSH targets. Refer to the following:

  - Chapter 6, "Working with Targets" for more information about configuring Network Devices such as routers, firewalls, and hypervisors like Oracle Virtual Machine through the SSH targets.
  - Appendix C, "Working with the SSH Connector" for more information configuring the SSH Connector.
- Added support for Copying Passwords to the Clipboard. Refer to Section 9.5.1, "Checking Out Passwords" for more information about how a user can copy the passwords that are provided during password checkout directly to the clipboard.
- Added support for the Clear Clipboard feature. Refer to Section 9.5.2, "Clearing Copied Passwords From the Clipboard" for more information about clearing the clipboard after a copied password has been used and is no longer needed for further use.
- Enhanced the Oracle Privileged Account Manager Usage Policies. Refer to Section 10.3, "Working with Usage Policies" for information about how to enable administrators to control users' session access and capabilities.
- Added the ability for administrators to delegate their administrative privileges using resource groups. Refer to Chapter 12, "Working with Resource Groups" for more information about managing resource groups and delegating administration.
- Enhanced the Oracle Privileged Account Manager plug-in framework. Refer to Chapter 13, "Working with Plug-Ins" for more information about how the following features enhance functionality, manageability, and fault tolerance:
  - Provided additional filtering rules to help manage plug-in implementations.
  - Enabled retry support for post plug-ins to provide fault tolerance.
  - Added a "create-like" feature and improved defining required attributes and defaults to better facilitate plug-ins.
- Enhanced various Oracle Privileged Account Manager reports. Refer to Chapter 15, "Working with Reports" for more information about improved user interfaces, additional metrics, and the new password age search option.
- The following common problems and their workarounds have been added to Chapter 20, "Troubleshooting Oracle Privileged Account Manager":

  Section 20.3.16, "End User Names Created in Oracle Identity Manager with the "#" Character Cannot Login to Oracle Privileged Account Manager"

  Section 20.3.17, "Audit Records Appear in BI Reports After a Long Delay"

  Section 20.3.18, "The "Failure to Load Windows Connector" Exception Occurs"

  Section 20.3.19, "Failure to Add a UNIX Target or Checkout a UNIX Account"

  Section 20.3.20, "Copying Password to Clipboard Fails in a HA Environment"

  Section 20.3.21, "Error in Loading SAP Classes During the Startup of the Server"

  Section 20.3.22, "Checkout History Search Results for Pattern Search Do Not Include Recent Session Recordings"

  Section 20.3.23, "The OPAMAgentService Windows Service Stops"

  Section 20.3.24, "A User is Able to Access the Grants of Another User"
- Added information about frequently asked questions. Refer to Section 20.4, "Frequently Asked Questions" for more information.

Various other changes have been made to the Console, command line, and RESTful interfaces. Information about these new or updated interface changes is provided throughout this guide.

New and Changed Features for 11*g* Release 2 (11.1.2.2.0)

Oracle Privileged Account Manager 11*g* Release 2 (11.1.2.2.0) included the following new and changed administrative, development, and security features:

- Added a plug-in framework that enables you to extend and customize Oracle Privileged Account Manager functionality to better suit your specific requirements. This framework enables you to
  - Validate and manipulate data before Oracle Privileged Account Manager performs operations
  - Perform specific actions after Oracle Privileged Account Manager completes its operations
  - Register and manage plug-ins through the Oracle Privileged Account Manager Console, command line, or RESTful interface
  - Integrate Oracle Privileged Account Manager with third-party systems such as wallets, ticket management systems, and audit systems

  In addition, a new Plug-in Configuration page and several new plug-in related options have been added to the Oracle Privileged Account Manager Console, command line tool, and RESTful interface. For more information about plug-ins and using the new interface features to configure plug-ins, refer to the following:
  - Chapter 1, "Introduction to Oracle Privileged Account Manager" for information about how the plug-in framework functionality works within Oracle Privileged Account Manager.
  - Section 2.6, "Understanding Plug-In Security" for information about plug-in security.
  - Chapter 4, "Starting and Using the Oracle Privileged Account Manager Console" for information about the using the Console features to search for and configure plug-ins.
  - Chapter 13, "Working with Plug-Ins" for basic information about configuring and deploying plug-ins in Oracle Privileged Account Manager by using the Console.
  - Chapter 18, "Developing Plug-Ins for Oracle Privileged Account Manager" for information about creating your own custom plug-ins for Oracle Privileged Account Manager.
  - Section A.9, "Working with Plug-Ins" for information about configuring and deploying plug-ins by using the command line tool.
  - Section B.12, "Plug-In Resource" for information about configuring and deploying plug-ins by using Oracle Privileged Account Manager's RESTful interface.
- Added the Oracle Privileged Session Manager to manage the privileged sessions to the target system. By creating a single access point to the target resources, Oracle Privileged Session Manager (Session Manager) helps administrators easily control and monitor all of the activities within the privileged session.

  In addition, a new Session Management page and several new session management-related updates have been made to the Console, command line tool,

and RESTful interface. For more information about session management and configuring sessions, refer to the following:

- Chapter 1, "Introduction to Oracle Privileged Account Manager"for information about how the Session Manager functionality works within Oracle Privileged Account Manager.
- Section 2.5, "Understanding Session Management Security" for information about Session Manager security.
- Section 5.3, "Managing the Oracle Privileged Session Manager Server" for information about configuring the Session Manager server.
- Chapter 9, "Working with Privileged Accounts" for information about administering managed sessions from the Console and about privileged sessions.
- Section 10.3, "Working with Usage Policies" for information about configuring sessions in Usage Polices.
- Section 14.7, "Checking Out Privileged Account Sessions" for information about how to check out sessions and passwords.
- Section 15.5, "Working with Checkout History Reports" for information about working with Session History reports.
- Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface" for information about administering managed sessions by using Oracle Privileged Account Manager's RESTful interface.

- Added a new `opsmconfig` configuration object that represents the configuration information for Session Manager servers.
- Added a new My Checkouts page where users can access a list of the accounts they currently have checked out and a Checkout History page where administrators can access information about account checkouts.
  - For information about the My Checkouts page, refer to Section 4.3, "Navigating Oracle Privileged Account Manager's Console" and Section 9.5, "Checking Out Privileged Accounts."
  - For information about the Checkout History page, refer to Section 15.5, "Working with Checkout History Reports."
  - Added new Checkout History Report that enables administrators to view information about any account checkouts performed over a specified period of time. Refer to Section 15.5, "Working with Checkout History Reports" for information.

Various other changes have been made to the Console, command line, and RESTful interfaces. Information about these new or updated interface changes is provided throughout this guide.

Other Significant Changes in This Guide

This section describes significant changes that have been made in this guide. The topics include:

- Significant Changes for Release 11gR2 (11.1.2.3.0)
- Significant Changes for Release 11gR2 (11.1.2.2.0)

Significant Changes for Release 11*g*R2 (11.1.2.3.0)

The following significant changes have been made in revision "2" of the guide:

- Section 5.4.1.1, "Installing and Configuring The Connector Server" has been modified.
- "ActiveDirectorySwitch" has been renamed to "WindowsLocalSwitch" in step 2 of Section 5.4.1.4, "Enabling Logging."
- "Administrators Account" row of Table 6–8, " Basic Configuration Parameters for the Windows Target Type" has been updated.

The following significant changes have been made in Release 11gR2 (11.1.2.3.0) of the guide:

Added and updated various parameter labels, procedure descriptions, and screenshots throughout this guide based on changes to the user interface, command line tool commands, and RESTful APIs.

Significant Changes for Release 11*g*R2 (11.1.2.2.0)

For release 11.1.2.2.0, this guide was reorganized and updated as follows:

- Added and updated various parameter labels, procedure descriptions, and screenshots throughout this guide based on changes to the user interface, command line tool commands, and RESTful APIs.
- Reorganized Chapter 5, "Configuring and Managing Oracle Privileged Account Manager," into smaller, separate chapters. Refer to the Contents for more information.
- Added the following new chapters and appendixes:
  - Chapter 4, "Starting and Using the Oracle Privileged Account Manager Console," describes how to invoke and work with Oracle Privileged Account Manager's web-based graphical user interface, or *Console*.
  - Chapter 7, "Working with Service Accounts" describes how to configure and manage OPAM Service Accounts.
  - Chapter 13, "Working with Plug-Ins," describes how to configure and deploy an Oracle Privileged Account Manager plug-in.
  - Chapter 18, "Developing Plug-Ins for Oracle Privileged Account Manager," describes how to write your own, custom plug-ins.
- Reorganized Chapter A, "Working with the Command Line Tool," by combining related commands into sections. For example, all of the server related- commands are now located in Section A.2, "Working with the Server." Refer to the Contents for more information.

Part I

Introduction to Oracle Privileged Account Manager

This part contains introductory and conceptual information about Oracle Privileged Account Manager, and it includes the following chapters:

- Introduction to Oracle Privileged Account Manager
- Understanding Oracle Privileged Account Manager Security

1

Introduction to Oracle Privileged Account Manager

This chapter introduces you to Oracle Privileged Account Manager by describing key concepts, features, and functionality.

This chapter includes the following sections:

1.1 What is Oracle Privileged Account Manager?

This section also discusses the following topics:

Oracle Privileged Account Manager manages *privileged accounts* that are not being managed by any other Oracle Identity Management components.

Accounts are considered "privileged," if they can access sensitive data, can grant access to sensitive data, or can both access and grant access to that data. Privileged accounts are your company's most powerful accounts and they are frequently shared.

Accounts become candidates for management via Oracle Privileged Account Manager if they are associated with elevated privileges, are used by multiple end-users on a task-by-task basis, and must be controlled and audited.

For example, the following accounts require security and may fall under compliance regulations:

- UNIX root, Windows administrator, and Oracle Database system accounts
- Application accounts, such as the database user accounts used by an application server when it connects to a Human Resources application

- Traditional shared and elevated privilege user accounts, such as system administrators and database administrators

> **Note:** Administrators determine which accounts are privileged within a particular deployment, and they must configure Oracle Privileged Account Manager to manage those accounts.

While Oracle Privileged Account Manager most commonly manages shared and elevated privileged accounts, administrators can also use it to manage passwords for any type of account. For example, if an employee is on extended leave and you have a business reason for allowing another employee to access the system using that person's email account, Oracle Privileged Account Manager can manage that privilege.

1.2 Why Use Oracle Privileged Account Manager?

Oracle Privileged Account Manager enables you to administer and provide better security for privileged accounts and passwords that are traditionally difficult to manage for several reasons.

First, privileged accounts generally have more access rights than a regular user's account. Because these accounts are not typically associated with one specific employee, they are often difficult to audit with existing tools and processes. Consequently, when employees leave the company, they might retain privileged account passwords that are still in use, which is a very serious compliance and security issue.

Also, changing privileged account passwords on a regular basis is difficult. If many people depend on the account, changing the password and notifying everyone requires a coordinated effort.

Finally, you typically do not want to store passwords in a central or well-known location, such as an external repository (like LDAP) or in application configuration files, because you cannot control access to those passwords.

Oracle Privileged Account Manager delivers a complete solution for securely managing privileged accounts and passwords because it provides the following features:

- Centralized password management for privileged and shared accounts, including UNIX and Linux root accounts, Windows accounts, Oracle Database SYS, application accounts, and LDAP admin accounts.
- Interactive, policy-based account and session *checkout* and *check-in*.

  Oracle Privileged Account Manager requires all authorized users to check out an account before using it, and then to check that account back in when they are finished with it. Oracle Privileged Account Manager audits account check outs and check ins by tracking the real identity (the person's name) of every shared administrator user at any given moment in time. By using this information, Oracle Privileged Account Manager can provide a complete audit trail that shows who accessed what, when, and where.

  In addition, Oracle Privileged Session Manager (Session Manager) enables administrators to monitor and control which activities users can perform during a session. Users are never allowed direct access to resources or to privileged credentials.
- Automatic password changes using the Identity Connector Framework (ICF).

Oracle Privileged Account Manager modifies passwords when they are checked out and checked in (when configured to do so). Consequently, when a user checks out a password and then subsequently checks it back in, that user can no longer use the previously checked out password.

In addition, Oracle Privileged Account Manager can change application privileged account passwords at specified intervals, such as every 90 days, with no changes to those applications and Oracle Privileged Account Manager synchronizes those passwords on the *target systems* (software systems that contain, use, and rely on user, system, or application accounts). For example, Oracle Privileged Account Manager can update service and scheduled task credentials.

- User management, group management, and workflow capabilities through Oracle Identity Governance Platform.

  Because Oracle Privileged Account Manager seamlessly integrates with Oracle Identity Manager, Oracle Privileged Account Manager can use this Oracle Identity Management product to manage the users and groups that are associated with a company's privileged accounts. In addition, through the request-level approval workflows, operational-level approval workflows, and provisioning workflows of Oracle Identity Manager, you can configure Oracle Privileged Account Manager so that only the appropriate groups and users have access to privileged accounts.

1.2.1 Features

Oracle Privileged Account Manager's key features include the following:

- Multiple access points such as:
  - Oracle Privileged Account Manager's web-based user interface (called the *Console*)

    The two following interfaces are associated with the Console:

    - Administrator: Oracle Privileged Account Manager administrators use this interface to create and manage policies, targets, accounts, grants, and reports.

    - Self-Service: Oracle Privileged Account Manager end users use this interface to search for, view, check out, and check in accounts.

    Refer to Chapter 4, "Starting and Using the Oracle Privileged Account Manager Console" for more information.

  - Oracle Privileged Account Manager's command line tool (CLI)

    You can use the CLI to perform many of the same tasks you perform from the Console. For example, you can use the CLI to check out and check in accounts or to create and manage policies, targets, accounts, and grants.

    Refer to Appendix A, "Working with the Command Line Tool" for more information.

  - RESTful APIs

    Oracle Privileged Account Manager uses RESTful APIs to expose internal functionality to applications and scripts. These APIs also provide the integration point to be leveraged by third parties that want to integrate with Oracle Privileged Account Manager functionality.

> **Note:** These APIs are considered to be RESTful because they conform to Representative State Transfer (REST) standards.
>
> Refer to Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface" for more information.

- Integration with Oracle technologies such as:
  - **Oracle Platform Security Services (OPSS) Policy Store**

    Oracle Privileged Account Manager integrates with Oracle Platform Security Services (OPSS) Policy Store for authorization.
  - **Oracle Platform Security Services (OPSS) Trust Service**

    Oracle Privileged Account Manager integrates with Oracle Platform Security Services (OPSS) Trust Service to authenticate and propagate identities from the Oracle Privileged Account Manager user interface to the Oracle Privileged Account Manager server
  - **Identity Connector Framework (ICF)**

    Oracle Privileged Account Manager uses the Identity Connector Framework (ICF) to connect to target systems and to discover, update, or discover and update the passwords for privileged accounts on those systems

    In addition, because ICF is an open standard, you can write your own connectors against other types of targets for which Oracle has not yet created an ICF connector.

    For more information about ICF and about developing your own connector, refer to "Understanding the Identity Connector Framework" and "Developing Identity Connectors Using Java" or "Developing Identity Connectors Using .Net" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.
- Ability to manage and audit privileged sessions to the target system as explained by the following examples:
  - Session Manager creates a single access point to target resources, which enables administrators to easily control and monitor all the activities within the privileged session.
  - Session Manager also maintains historical records (transcripts) to support forensic analysis and audit data.
- Support for multiple target types such as:
  - UNIX and Linux operating systems
  - Windows operating systems
  - Oracle, MSSQL, MySQL and Sybase databases
  - LDAP v3-compliant directories
  - Hypervisors
  - Network devices
- Advanced reporting capabilities such as:
  - Oracle Privileged Account Manager's out-of-the box audit reports are integrated with Oracle Business Intelligence Publisher 11g (BI Publisher) so

you know who is using your privileged accounts. BI Publisher also enables you to create and manage formatted reports from different data sources.

  - The Oracle Fusion Middleware Audit Framework logs audit events in a centralized database. Oracle Privileged Account Manager uses these events to generate audit reports.
  - Events related to privileged account access roll up into Oracle Identity Manager and Oracle Identity Analytics for audit and attestation.

- Policy-driven access to privileged accounts as explained by the following examples:

  In Oracle Privileged Account Manager, there are two types of policies for granting access to privileged accounts:

  - Password Policy

    This policy type captures the password construction rules enforced by a specific target on an associated privileged account. For example, you can specify the minimum and maximum number of numeric characters for a password for an account. In addition, you use a password policy to create a password value that Oracle Privileged Account Manager uses to reset a password for a privileged account.

  - Usage Policy

    This policy type defines when and how often a user or group can access a privileged account.

    Note: If you do not specify a time interval by using a Usage Policy, the user or group can access the privileged account at any time.

- Ability to manage *attended* and *unattended* accounts as explained in the following descriptions:

  - An attended account is an account assigned to a particular group or user.
  - An unattended account is an account that is never used by an end user.

    For example, Oracle Privileged Account Manager uses an unattended account, called the *OPAM service account*, to connect to and manage target systems. This account performs all Oracle Privileged Account Manager-related operations (such as discovering accounts, resetting passwords, and so forth) on the target system, which is why the OPAM service account (service account) must have some special privileges and properties.

    Oracle Privileged Account Manager can also manage other kinds of unmanaged accounts, such as an application account or a service account with Credential Store Framework (CSF) mappings that enable applications to pick up a password at run-time by using CSF.

    Note: You must never use the same account as a service account *and* a privileged account to be managed by Oracle Privileged Account Manager.

    For more information about working with service accounts in Oracle Privileged Account Manager, refer to Section 7, "Working with Service Accounts."

1.2.2 Functionality

In addition to the functionalities described in Section 1.2, "Why Use Oracle Privileged Account Manager?," Oracle Privileged Account Manager also performs the following functions:

- Associates privileged accounts with targets.
- Grants users and roles access to privileged accounts, and removes that access.
- Provides an extensible plug-in framework that enables you to use Oracle or third-party plug-ins to perform operations such as custom notifications, extended usage policies, and custom logic to synchronize passwords with external repositories.
- Provides role-based access to accounts maintained in the Oracle Privileged Account Manager accounts request system.
- Provides password check out and check in, as well as session checkout to control access to accounts.
- Provides "over-the-shoulder" session management by enabling administrators to perform the following actions:
  - Control session initiation
  - Control sessions through policy-based and administrator-initiated session termination and lockout
  - Monitor and audit sessions
- Eliminates the potential of having unmanaged privileged accounts when your unattended applications use client-certificate authentication

  Client-certificate authentication is performed using an SSL certificate to authenticate (in lieu of a password) against an Oracle Privileged Account Manager server.
- Resets passwords to a random value on check in and check out by default

  You can configure Oracle Privileged Account Manager to automatically check in privileged accounts after a specified time to protect against users who check out that privileged account and do not bother to explicitly check in the account.

  You can also control the duration in which users can keep a privileged account checked out.
- Manages password resets on supported targets.
- Makes authorization decisions to determine the following conditions:
  - Which targets, privileged accounts, and policies are exposed to an end user or administrator.
  - Which operations (such as add, modify, check-in, and checkout) end users and administrators can perform.
- Associates policies with privileged accounts.
- Performs and supports Create, Read, Update, Delete, and Search (CRUDS) operations on targets, privileged accounts, and policies.

  This core functionality is exposed through Oracle Privileged Account Manager's RESTful APIs. Check-ins, checkouts, and so forth are also supported through the RESTful interface.

- Uses Oracle's common auditing, logging, and reporting capabilities to monitor and report access.

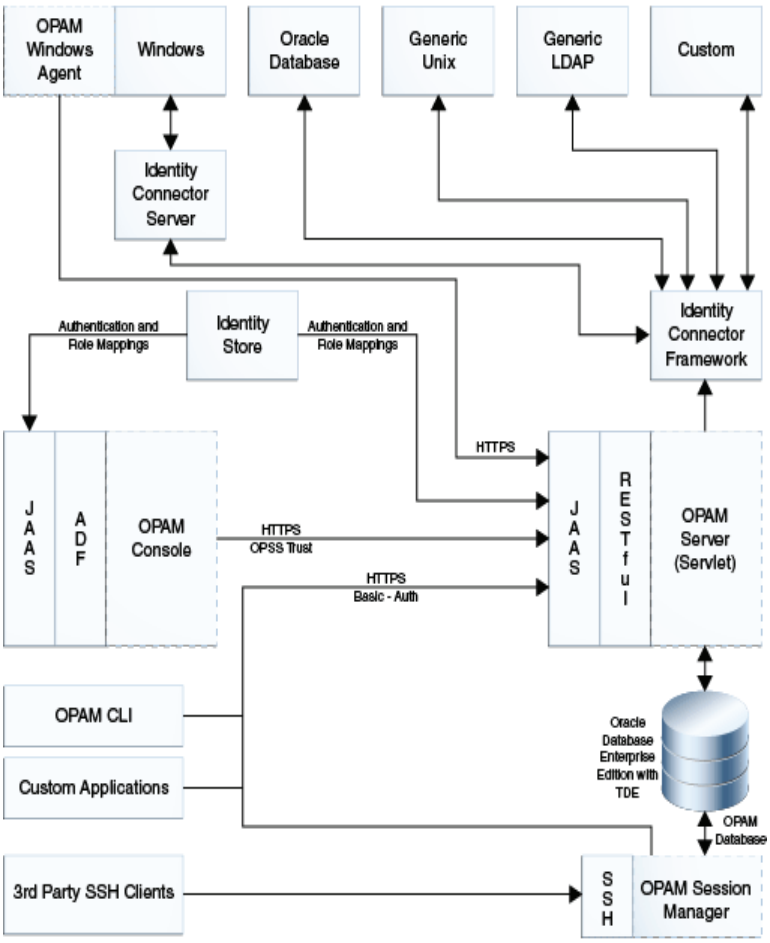
  With Oracle Privileged Account Manager, you can use the auditing, logging, and reporting capabilities of Oracle Fusion Middleware Control and Oracle BI Publisher to monitor and report access that users and groups have to privileged accounts.

- Offers multiple high availability capabilities.

1.2.3 Architecture and Topology

The following diagram illustrates the architecture and topology of Oracle Privileged Account Manager:

Figure 1–1 Oracle Privileged Account Manager Architecture and Topology



As you examine this figure, it is important to note the following points:

- All of Oracle Privileged Account Manager's core logic resides on the Oracle Privileged Account Manager server. This functionality is exposed through a Representational State Transfer (REST or RESTful) service, where the data is encoded as JavaScript Object Notation (JSON).

> **Note:** Oracle Privileged Account Manager provides a web-based user interface (known as the *Console*) and an Oracle Privileged Account Manager command line tool (CLI). Both interfaces are essentially clients of the Oracle Privileged Account Manager server.
>
> However, third parties can write their own clients, such as custom applications, by leveraging the open RESTful service. For more information, refer to Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface."

- Session Manager is an Oracle Privileged Account Manager subcomponent that empowers Oracle Privileged Account Manager's session management capabilities. Session Manager is a J2EE application that interacts with the Oracle Privileged Account Manager Server through the Oracle Privileged Account Manager RESTful interfaces and shares the same database that is used by the Oracle Privileged Account Manager Server. In addition, the Session Manager listens and responds to SSH traffic to establish privileged sessions against SSH-capable Oracle Privileged Account Manager targets.
- Oracle Privileged Account Manager authentication relies on Java Authentication & Authorization Service (JAAS) support in the J2EE container on which its deployed.

  Refer to "WebLogic Security Service Architecture" in *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server* for more information about JAAS support in Oracle WebLogic Server (WebLogic).

  For more information about Oracle Privileged Account Manager authentication, refer to Section 2.2, "Understanding Oracle Privileged Account Manager Authentication."
- All communication with, and between, Oracle Privileged Account Manager-related components (including Oracle Privileged Account Manager's Console, command-line interface, and server) occurs over SSL. In addition, Oracle Privileged Account Manager's RESTful interfaces are exposed over SSL.
- Oracle Privileged Account Manager relies on and transparently uses the identity store, policy store, and credential store configured for the WebLogic domain in which Oracle Privileged Account Manager is deployed. Because the policy store and credential store are implicitly part of the WebLogic domain, they are not depicted in this diagram.

  The identity store is the centralized repository for Oracle Privileged Account Manager users and groups.

  - The OPSS identity store can point to the LDAP embedded in WebLogic (out of the box) or to an external LDAP server.

    Refer to "Configuring the Identity Store Service" in the *Oracle Fusion Middleware Application Security Guide* for configuration instructions.
  - For information about managing the policy store and the credential store, refer to "Managing the Policy Store" and "Managing the Credential Store" in the *Oracle Fusion Middleware Application Security Guide*.
- The Oracle Privileged Account Manager Console leverages, and is rendered by, Oracle Application Development Framework (**ADF**).

  For more information about ADF, refer to the following website:

http://www.oracle.com/technetwork/developer-tools/adf/overview/index.html

- Oracle Privileged Account Manager connects to targets by using Identity Connector Framework (ICF) connectors. As shown in Figure 1–1, Oracle Privileged Account Manager uses the following connectors, which are constructed by using the ICF:
  - Generic Database User Management Connector

    This connector connects to Oracle, MSSQL, Sybase, MySQL databases.
  - Generic Unix Connector

    This connector connects to any UNIX system.
  - Generic LDAP Connector

    This connector connects to LDAP targets (such as Oracle Internet Directory, Oracle Universal Directory, and Active Directory).
  - Custom Connector

    This connector connects to a target that does not have a predefined connector associated with it.
  - SSH Connector

    This connector connects to Network Devices.
  - SAP Connector

    This connector connects to the SAPUM and SAPUME targets.
  - Windows Connector

    This connector connects to Microsoft Windows targets.

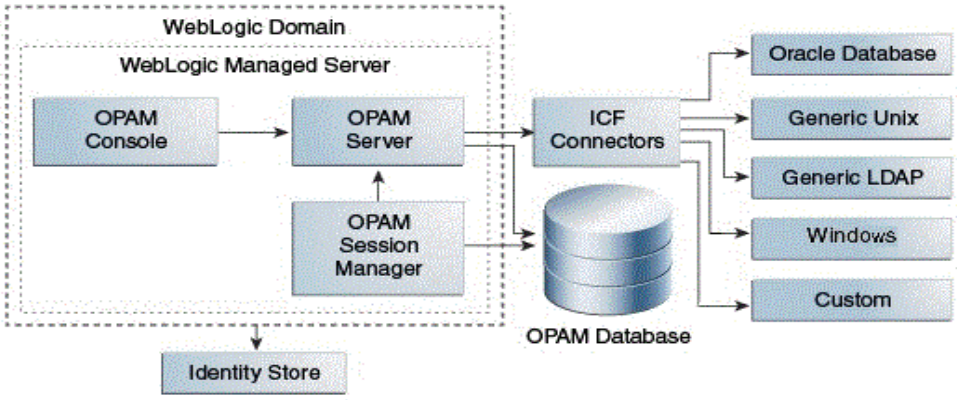
  For additional information, refer to "Understanding the Identity Connector Framework" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.

1.3 How Oracle Privileged Account Manager is Deployed in Oracle Fusion Middleware

The following figure illustrates how Oracle Privileged Account Manager is deployed within Oracle Fusion Middleware.

Note: If you are using Oracle Privileged Account Manager on IBM WebSphere, refer to "Differences in How Oracle Privileged Account Manager is Deployed in Oracle Fusion Middleware" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for information about this topic.

Figure 1–2 Oracle Privileged Account Manager Deployed Within Oracle Fusion Middleware



As you examine this figure, note the following points:

- All components are deployed within a single WebLogic domain.
- Oracle Privileged Account Manager stores its application data in the Oracle Privileged Account Manager database. In addition, the Oracle Privileged Account Manager schema is created in this database via the Oracle Repository Creation Utility.
- Oracle Privileged Session Manager relies on the Oracle Privileged Account Manager Database for persistence and communicates with Oracle Privileged Account Manager through its RESTful interfaces.
- Oracle Privileged Account Manager's web-based user interface (the Console) is deployed in the Oracle WebLogic Server Managed Server, along with the Oracle Privileged Account Manager Server and the Session Manager.

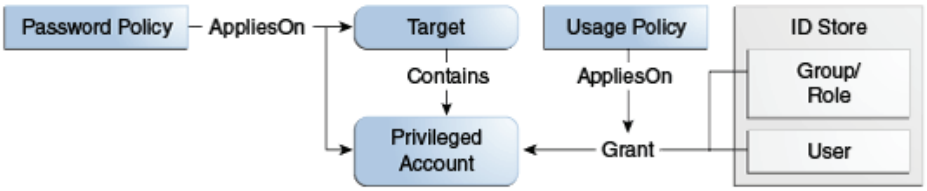
  The Console communicates with the Oracle Privileged Account Manager Server. This server is created as a server that is managed by the Oracle WebLogic Server Managed Server (or Managed Server).
- The OPSS identity store and the OPSS security store (which includes the policy store and credential store) are WebLogic domain-wide constructs, so there is one of each per domain. Because the OPSS security store is implicitly part of the WebLogic domain, it is not depicted in this diagram.

  Oracle Privileged Account Manager simply works with what is configured for that domain. You are not required to use an Oracle Privileged Account Manager-specific configuration to use these constructs and services. In addition, Oracle Privileged Account Manager abstracts out the use of these constructs and services so that you do not have to understand the background functionality in great detail.

1.4 Understanding the Relationship between Oracle Privileged Account Manager Entities

Before you start working with the different Oracle Privileged Account Manager entities, you should understand how those entities relate to each other. Figure 1–3 illustrates this relationship.

Figure 1–3 Oracle Privileged Account Manager Entity Relationships



An Oracle Privileged Account Manager Password Policy can apply on both a target or a privileged account. When applied on a privileged account, that account's password construction (its complexity) and lifecycle (how often it changes) is governed by the effective Oracle Privileged Account Manager Password Policy. Similarly, when applied on a target, the target's service account is governed by the Oracle Privileged Account Manager Password Policy.

Targets are software systems that contain one or more privileged accounts.

A Usage Policy applies on a grant and it controls when and how grantees can use a privileged account. For example, you can configure a Usage Policy to control when a user's access to an account will expire.

Users and groups (roles) are maintained in the Oracle Privileged Account Manager identity store. These users and groups can only access a privileged account through a grant. If a user or group member tries to access a privileged account, and Oracle Privileged Account Manager finds a grant, then the grantee is allowed to access the account based on that grant and its associated Usage Policy.

1.5 System Requirements and Certification

Refer to the system requirements and certification documentation for information about hardware and software requirements, platforms, databases, and other information. Both of these documents are available on Oracle Technology Network (OTN).

The system requirements document covers information such as hardware and software requirements, minimum disk space and memory requirements, and required system libraries, packages, or patches:

http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-requirements-100147.html

The certification document covers supported installation types, platforms, operating systems, databases, JDKs, and third-party products:

http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html

2

Understanding Oracle Privileged Account Manager Security

This chapter describes how Oracle Privileged Account Manager authenticates and authorizes different types of users by using the authentication and authorization framework provided in the Oracle Privileged Account Manager server. In addition, this chapter explains various methods that you can use to further secure Oracle Privileged Account Manager in your deployment environment.

This chapter includes the following sections:

2.1 Overview

The authentication and authorization framework provided in the Oracle Privileged Account Manager server provides the following features and functionality:

- Supports OPSS-Trust tokens and HTTP-Basic Authentication.

  You can also configure the Oracle Privileged Account Manager Console to work alongside Oracle Access Management. Refer to Section 19.2, "Integrating with Oracle Access Management Access Manager" for more information.

- Leverages the Java Authentication & Authorization Service (JAAS) for authentication.

  Note: Oracle Privileged Account Manager authentication relies on JAAS support in the J2EE container on which its deployed. Refer to "WebLogic Security Service Architecture" in *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server* for more information.

- Defines different Oracle Privileged Account Manager-specific Admin Roles and their Oracle Privileged Account Manager-specific responsibilities.

- Enforces authorization decisions that determine the following conditions:

  - Which targets and privileged accounts are exposed to an administrator or to an end-user.
  - Which operations (such as add, modify, check-in, and checkout) an end-user or an administrator can perform on targets, privileged accounts, and policies.
- Supports Usage Policies and Password Policies for privileged accounts

2.2 Understanding Oracle Privileged Account Manager Authentication

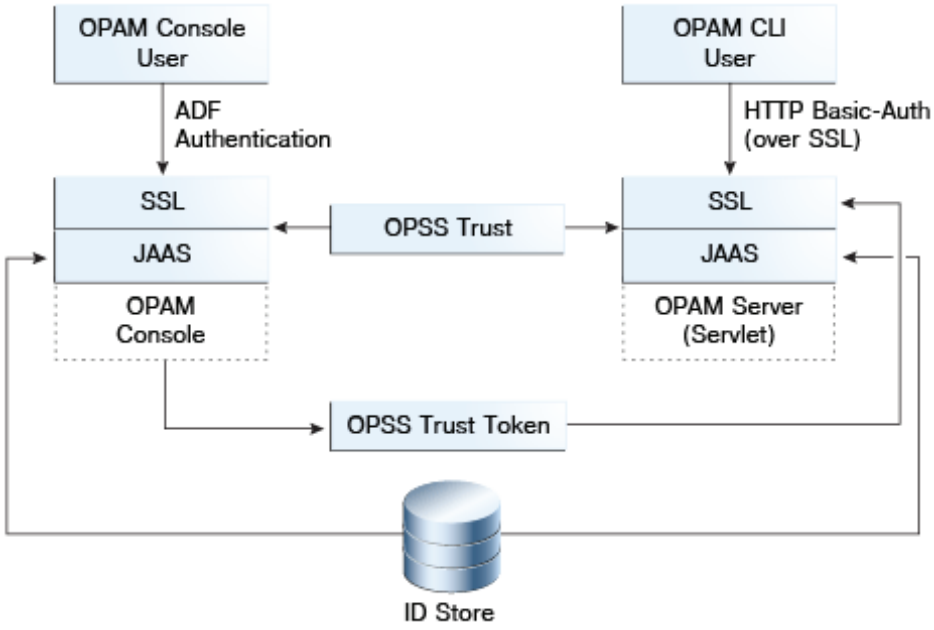
This section also discusses the following topics:

- Section 2.2.1, "Authentication for the Oracle Privileged Account Manager Console"
- Section 2.2.2, "Authentication for the Oracle Privileged Account Manager Server"

Oracle Privileged Account Manager uses Security Assertions Markup Language-based (SAML-based) tokens as its authentication mechanism. SAML-based token authentication is provided by using the OPSS Trust Service in the J2EE container on which its deployed.

The following figure illustrates Oracle Privileged Account Manager authentication.

Figure 2–1 Trust-Based Authentication in Oracle Privileged Account Manager



Trust Service instances are typically configured to securely propagate user identities from the client application to the Oracle Privileged Account Manager server as part of the Oracle Privileged Account Manager installation and configuration process.

Oracle Privileged Account Manager requires authentication under the following conditions:

- When users interact with Oracle Privileged Account Manager's web-based user interface (the Console) and the command line tool (CLI).
- When users and clients interact directly with the Oracle Privileged Account Manager server through its RESTful interfaces.

In both cases, Oracle Privileged Account Manager supports the following predefined authentication modes, over SSL:

- **HTTP Basic-Authentication**: The user sends the user name and password as unencrypted base64 encoded text.
- **OPSS-Trust Service Assertions**: The OPSS Trust Service allows the propagation of identities across HTTP-enabled applications by providing and validating tokens. This service uses an asserter that is available through Oracle WebLogic Server.

In addition, Oracle Privileged Account Manager can support ADF-based authentication for UI-based interactions, which is done transparently against the domain-specific identity store.

2.2.1 Authentication for the Oracle Privileged Account Manager Console

The Oracle Privileged Account Manager web-based user interface, or *Console*, supports ADF-based authentication mechanisms and you can configure the interface with Oracle Access Management.

When a user interacts with the Oracle Privileged Account Manager Console, the following occurs:

1. The user authenticates against the Oracle Privileged Account Manager Console by using ADF authentication.
2. The Oracle Privileged Account Manager Console calls the OPSS-Trust Service to request a token that asserts the identity of the user logged into the Oracle Privileged Account Manager Console.
3. Now, whenever the Oracle Privileged Account Manager Console makes RESTful calls to the Oracle Privileged Account Manager server to execute Oracle Privileged Account Manager functionality, the Oracle Privileged Account Manager Console presents the generated token to the Oracle Privileged Account Manager server.
4. Because the OPSS Trust Service Asserter is configured by default, the Asserter examines the token presented in the previous step, validates the token, and then asserts that the identity performing the RESTful call against the Oracle Privileged Account Manager server is the one contained in the token.

   This process is called *identity propagation*. An end-user only authenticates against the Oracle Privileged Account Manager Console, but the Console can securely convey to the Oracle Privileged Account Manager server the identity for which they are making a request.

   The important point to note about identity propagation is that it removes the need for end users to authenticate themselves against the Oracle Privileged Account Manager Console and the Oracle Privileged Account Manager server.

> **Note:** If you deploy your own client applications against the Oracle Privileged Account Manager server, then you must have identity propagation. In such a context, it is recommended that you use OPSS-Trust Service based Identity Assertions. For more information, refer to the *Oracle Fusion Middleware Application Security Guide*.

2.2.2 Authentication for the Oracle Privileged Account Manager Server

The Oracle Privileged Account Manager server only exposes RESTful interfaces and supports HTTP-Basic Authorization or OPSS-Trust. In addition, the Oracle Privileged

Account Manager server requires that all communication with that server occurs over an SSL-secured channel.

The Oracle Privileged Account Manager command line tool client uses HTTP Basic-Authentication over SSL to connect to, and authenticate against, the Oracle Privileged Account Manager server.

2.3 Understanding Oracle Privileged Account Manager Authorization

This section describes Oracle Privileged Account Manager authorization.

The topics include:

- Section 2.3.1, "Administration Role Types"
- Section 2.3.2, "End Users"

> **Note:** If you are using Oracle Privileged Account Manager on IBM WebSphere, refer to "Differences in Oracle Privileged Account Manager Authorization" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for information about this topic.

2.3.1 Administration Role Types

Common Admin Roles are a set of predefined, standardized application roles for securing administrative access to Oracle Identity Management applications. These roles encapsulate the common administrative tasks across the Oracle Identity Management suite.

Oracle Privileged Account Manager uses Admin Roles to manage access to targets and privileged accounts and to control which operations administrators can perform. Specifically, the Oracle Privileged Account Manager server renders different user interface components based on the Admin Role assigned to the user logging in.

Only administrators who are assigned the Oracle Privileged Account Manager-specific Admin Roles can administer Oracle Privileged Account Manager.

Enterprise roles must be created in the domain identity store to support the Oracle Privileged Account Manager Admin Roles.

The following are the prerequisites to configuring enterprise roles for the Common Admin Roles:

- The domain identity store must be configured. For more information, refer to Section 3.3.2, "Configuring an External Identity Store for Oracle Privileged Account Manager" and Section 3.3.3, "Preparing the Identity Store."
- The domain policy store must be configured. For more information, refer to Oracle Fusion Middleware Application Security Guide.

For more information about supported identity and policy store configurations for Oracle Privileged Account Manager, refer to Section 3.1, "Before You Begin."

The following table describes the Common Admin Roles that are specific to Oracle Privileged Account Manager.

Table 2–1 Supported Admin Roles

Admin Role	Access Rights	Responsibility	Skills and Expertise Required
Application Configurator (`OPAM_APPLICATION_CONFIGURATOR`)	▪ Configure and manage Oracle Privileged Account Manager Console and servers. ▪ Manage plug-in configurations (create plug-in configurations, modify configuration attributes the plug-in needs to work, and delete the configurations). **Note**: When a new plug-in configuration is created, the status is disabled and the plug-in cannot be executed. This role cannot enable a plug-in configuration. Only the Security Administrator (OPAM_SECURITY_ADMIN) role can enable plug-in configurations and determine under which conditions those plug-ins can be executed.	Use Identity Management applications to support business requirements within an assigned business scope.	▪ Strong knowledge of product features. ▪ Good knowledge of business requirements.

Table 2–1 (Cont.) Supported Admin Roles

Admin Role	Access Rights	Responsibility	Skills and Expertise Required
Security Administrator (OPAM_SECURITY_ADMIN)	■ Manage accounts (add, edit, and remove accounts) and view the Password History for an account. **Note**: This role cannot assign grantees to privileged accounts. ■ Manage targets (add, edit, and remove targets) and view the Password History for a target. ■ Manage resource groups (create, edit, and remove resource groups and users or groups with delegated access privileges). ■ Manage Password and Usage Policies (create, edit, and delete policies). ■ Assign Password Policies to accounts. **Note**: This role cannot assign Usage Policies because they are always associated with a grant. Only the User Manager (OPAM_USER_MANAGER) role can assign grants and Usage Policies. ■ Edit plug-in configurations, enable or disable plug-ins, and configure Filter Rules (such as enabling users or groups) to decide under which conditions a plug-in can be executed. **Note**: This role cannot create or delete a plug-in configuration. Only the Application Configurator (OPAM_APPLICATION_CONFIGURATOR) role can create or delete a plug-in configuration. ■ Search for and edit Plug-In Configurations and Connector Server Configurations. ■ View properties on the Server Configuration and Session Manager Configuration pages.	■ Configure Identity Management application roles and approve role grants. ■ Configure Identity Management applications to work with corporate infrastructure and applications. ■ Maintain system credentials for identity stores, key stores, databases, and other repositories. ■ Grant administrative roles and permissions.	■ Strong knowledge of corporate infrastructure and Identity Management security architecture. ■ Strong technical knowledge to troubleshooting infrastructure access rights.
Security Auditor (OPAM_SECURITY_AUDITOR)	Open and review Oracle Privileged Account Manager reports.	■ Provide audit reports to upper management. ■ Verify permissions and generate access reports. ■ Verify proper configuration of Identity Management applications.	■ Strong knowledge of access management processes. ■ Strong knowledge of the risks associated with unauthorized access. ■ Good understanding of information security and system architecture.

Table 2–1 (Cont.) Supported Admin Roles

Admin Role	Access Rights	Responsibility	Skills and Expertise Required
User Manager (OPAM_USER_MANAGER)	■ Assign end users with grants to privileged accounts. ■ Manage Usage Policies (create, edit, and delete Usage Policies). ■ Assign Usage Policies to grants. **Note**: The relationship between an account and a grantee (end user) of that account is called a *grant*. The User Manager can assign different Usage Policies to different grantees of the same account. This role cannot assign Password Policies to accounts. ■ View the Plug-In Configuration page and search for plug-ins. ■ Terminate all Oracle Privileged Session Manager sessions for a selected account.	■ Create, modify, and delete users and groups. ■ Reset passwords and unlock accounts.	Strong knowledge of corporate identity infrastructure.

2.3.2 End Users

Oracle Privileged Account Manager End Users or Enterprise Users are not assigned any roles, so they have limited access to Oracle Privileged Account Manager user interface components. These users are only entitled to perform certain tasks; which includes viewing, searching, checking out, and checking in privileged accounts for which they have been granted access.

Note: Refer to Chapter 14, "Working with Self-Service" for more information.

2.4 Securing Oracle Privileged Account Manager

You can implement the recommendations described in this section to further secure Oracle Privileged Account Manager in your deployment environment.

This section includes the following topics:

- Section 2.4.1, "Securing the Network Channel"
- Section 2.4.2, "Securing Shared Accounts"
- Section 2.4.3, "Enabling Password Resets"
- Section 2.4.4, "Avoiding Assignments through Multiple Paths"
- Section 2.4.5, "Defining Richer Password Policies"
- Section 2.4.6, "Delegating Administration"
- Section 2.4.7, "Hardening the Back-End Oracle Privileged Account Manager Database"

2.4.1 Securing the Network Channel

As part of its normal functionality, Oracle Privileged Account Manager performs remote password resets on target systems. Because these passwords allow access to

those systems as privileged identities (Oracle Privileged Account Manager manages privileged accounts and identities) you must ensure that these remote password resets occur over a secured network channel.

After being reset, Oracle Privileged Account Manager propagates these passwords to end users who are requesting access to the target system as a privileged account. Again, you must ensure that these newly reset passwords are propagated to the end users over a secured channel.

Considering these points, the two following aspects of an Oracle Privileged Account Manager deployment that must be closely examined and secured:

- Section 2.4.1.1, "Connecting to Target Systems"
- Section 2.4.1.2, "Securing the End User Interface"

2.4.1.1 Connecting to Target Systems

Oracle Privileged Account Manager leverages ICF connectors to communicate with target systems. These connectors are highly flexible and they can be configured in several ways. To allow flexibility in testing (and even production), Oracle Privileged Account Manager does not mandate that this connectivity always occurs over a secure channel.

Except for the Generic UNIX targets, which mandates SSH, the Generic LDAP and Generic DB targets allow connections through both secured (encrypted) and clear channels. Therefore, it is important for an Oracle Privileged Account Manager administrator to consider all relevant factors when deciding what type of channel to use when connecting to target systems.

Oracle recommends that you always use secured channels to mitigate the risk of password compromise due to packet sniffing. If the target system (either LDAP or DB) supports SSL and is listening on an SSL port, then Oracle Privileged Account Manager can communicate with that target over SSL.

Consult your target systems' product documentation for information about configuring your targets so that they are listening on an SSL port. To configure Oracle Privileged Account Manager to communicate through SSL, refer to Section 17.1, "Configuring Oracle Privileged Account Manager to Communicate With Target Systems Over SSL." Securing these connections through SSL ensures that the password reset operations performed by Oracle Privileged Account Manager occur in a secure manner.

2.4.1.2 Securing the End User Interface

The following are the two primary interfaces open to an Oracle Privileged Account Manager end user:

- Console (Refer to Chapter 4, "Starting and Using the Oracle Privileged Account Manager Console" for more information.)
- Command line tool (Refer to Appendix A, "Working with the Command Line Tool" for more information.)

Oracle Privileged Account Manager's Console can be deployed with SSL enabled or disabled.

If you deploy the Oracle Privileged Account Manager Console with SSL disabled, even if the Console communicates with the Oracle Privileged Account Manager server over an SSL secured channel, then the connectivity between the Console and the end user browser is not secured, which can cause security concerns.

Oracle recommends using the Oracle Privileged Account Manager Console SSL-enabled mode because it is more secure.

Because the Oracle Privileged Account Manager server mandates SSL connectivity, the Oracle Privileged Account Manager command line tool always uses SSL and communicates over a secure channel. Consequently, when the Oracle Privileged Account Manager server propagates a password to an end user through the command line tool, it always uses a secured channel and prevents compromises from packet sniffing.

2.4.2 Securing Shared Accounts

Oracle Privileged Account Manager enables you to specify whether a privileged account is *shared* or *not shared*. This section defines shared accounts, explains some security considerations, and describes how to improve security for a shared account.

This section also discusses the following topics:

- Section 2.4.2.1, "What is a Shared Account?"
- Section 2.4.2.2, "Security Limitations"
- Section 2.4.2.3, "How to Secure the Account"

2.4.2.1 What is a Shared Account?

By default, Oracle Privileged Account Manager allows only one user to check out an account at a time. If a second user tries to check out an already checked-out account, an error message is displayed stating the account is already checked out.

Oracle Privileged Account Manager also enables you to configure a *shared* account, which enables multiple users to check out the account at the same time.

When multiple users check out a shared account, Oracle Privileged Account Manager shares the password generated by the first user instead of generating a new password for each user. Setting a new password would affect the existing check out. Oracle Privileged Account Manager does not reset that password until all users have checked in the account and the last person has checked in the password.

Oracle recommends that you designate an account as shared only if there are compelling business reasons to do so. For example, sharing a database account might be advantageous if that account is being administered by multiple people.

2.4.2.2 Security Limitations

When you configure a shared account, keep in mind the following security limitations:

- Users can still use the password after checking in an account because Oracle Privileged Account Manager does not reset the password until the last user checks it in.
- Sharing accounts presents a problem with achieving a fine-grained audit. Oracle Privileged Account Manager can provide an audit trail that shows when the account was checked out and which users had access to that account at any given time. However, if multiple end users have the same privileged account checked out at the same time, then Oracle Privileged Account Manager cannot isolate the actions taken by an individual end user.

2.4.2.3 How to Secure the Account

If you do have a compelling reason for sharing an account, its useful to take the following steps to secure that account:

1. Configure the Usage Policy to automatically check in the privileged account after a specified period of time. Automatic check-ins ensure that shared privileged accounts get checked in and that passwords get cycled in a timely manner.
2. Limit the number of users to whom you assign the privileged account and try to further segregate these users by specifying when they can access the account. You can configure the Usage Policy to specify which days of the week and what times of the day a user can access an account. These limitations can minimize overlapping checkouts, which improves Oracle Privileged Account Manager's ability to audit.

> **Note:** For more information about configuring a Usage Policy, refer to Section 10.3.4, "Modifying the Default Usage Policy" or Section 10.3.5, "Creating a Usage Policy."

2.4.3 Enabling Password Resets

Oracle Privileged Account Manager allows you to configure the Password Policy for a privileged account so that Oracle Privileged Account Manager automatically resets the privileged account's password when the account is checked-out, checked-in, in both cases, or in neither case.

At a minimum, Oracle recommends that you configure and apply a Password Policy to reset the privileged account's password on check in. Resetting the password on check in prevents end users from using that account after checking it in because the password they used is no longer associated with that privileged account. This feature is one of the fundamental innovations in Oracle Privileged Account Manager and should be used.

> **Note:** For more information about configuring and working with Password Policies, refer to Chapter 10, "Working with Policies."

2.4.4 Avoiding Assignments through Multiple Paths

In addition to directly assigning privileged accounts to end users, Oracle Privileged Account Manager allows you to assign privileged accounts to groups. For example, you might want to create a "Data Center Product UNIX Administrators" group and give that group access to certain privileged accounts.

When designing your deployment, it is important to ensure that a given end user is granted access to a privileged account through only one path (either directly or through a single group). When Oracle Privileged Account Manager discovers multiple grant paths, it picks the first path retrieved from its back-end, which leads to non-deterministic behavior. This behavior can cause the *effective* Usage Policy to be different from the *intended* Usage Policy.

On a related note, you must avoid creating groups with multiple naming attribute values or you might enable users to access groups for which they were not explicitly granted access. Refer to Section 20.3.10, "Oracle Privileged Account Manager End Users Gain Privileges They Were Not Explicitly Granted" for more information.

> **Note:** For more information about configuring and working with grantees, refer to Chapter 11, "Working with Grantees."

2.4.5 Defining Richer Password Policies

The primary purpose of an Oracle Privileged Account Manager's Password Policy is to ensure the success of an Oracle Privileged Account Manager-initiated password reset that occurs against a target system.

At a minimum, Oracle Privileged Account Manager requires the effective Password Policy on a privileged account to describe the Password Policy being enforced on the target system. However, Oracle Privileged Account Manager administrators are not restricted to this requirement. You can define a much richer Password Policy in Oracle Privileged Account Manager that generates more complex and secure passwords during Oracle Privileged Account Manager reset operations.

Note: For more information about configuring and working with Password Policies, refer to Chapter 10, "Working with Policies."

2.4.6 Delegating Administration

Administrative privileges on some resources may need to be assigned or delegated to a particular user or role. In such cases, after the delegation of privileges from the administrator, the delegatee has delegated administration privileges on the assigned resource. However, the delegatee does not have any privileges on any of the other resources within the system. This is delegated administration.

For example, the security administrator may want to assign another user to administer a particular LDAP server. In this situation, the assigned user is not in the OPAM_SECURITY_ADMIN group of Oracle Privileged Account Manager, and so he does not have security administration privileges on all the other resources. However, he now have all the security administration privileges. In such a case, the security administration privileges for the LDAP resource are delegated to him. This is the delegation of the security administration privileges. Similarly, "User Management" Privileges can also be delegated.

2.4.7 Hardening the Back-End Oracle Privileged Account Manager Database

Beginning with the 11*g*R2 11.1.2.1.0 release, Oracle Privileged Account Manager moved to using the Oracle RDBMS as its data store. As such, it is important to ensure that the back-end Oracle Privileged Account Manager database is locked down for production deployments.

Oracle recommends that administrators perform the following three tasks to effectively lock down a back-end Oracle Privileged Account Manager database:

- Enable TDE
- Enable SSL
- Use Oracle Database Vault

Enable TDE

Oracle recommends enabling Transparent Data Encryption (TDE) mode on your back-end Oracle Privileged Account Manager database for all production deployments of Oracle Privileged Account Manager. Enabling TDE ensures that all sensitive information stored by Oracle Privileged Account Manager (such as account passwords) is encrypted on disk.

For security purposes, enabling TDE for Oracle Privileged Account Manager is a two-step process. You must

1. Enable TDE in the back-end database

   Refer to "Enabling TDE in the Database" in the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*.

2. Use any of the two following methods to configure Oracle Privileged Account Manager to expect a back-end database that is configured to use TDE:

   - To configure Oracle Privileged Account Manager from the Console, refer to Section 5.2.3, "Managing Oracle Privileged Account Manager Server Properties."
   - To configure Oracle Privileged Account Manager from the command line, refer to Section 17.2.1, "Enabling TDE Mode."

   Note: for more information about using Transparent Data Encryption, refer to the following websites:

   - To learn more about TDE best practices, refer to:

     `http://www.oracle.com/technetwork/database/security/twp-transparent-data-encryption-bes-130696.pdf`

   - To view the TDE FAQ, refer to:

     `http://www.oracle.com/technetwork/database/security/tde-faq-093689.html`

Enable SSL

In addition to the on-disk encryption provided by TDE, Oracle Privileged Account Manager obfuscates all sensitive information that it stores. This obfuscation ensures that anyone gaining access to the Oracle Privileged Account Manager schema still does not have direct access to sensitive information. Furthermore, the obfuscation prevents any malicious elements monitoring network traffic from easily viewing sensitive information. However, Oracle recommends that you perform *both* of the following actions for production deployments of Oracle Privileged Account Manager:

- Enable SSL on the back-end RDBMS used by Oracle Privileged Account Manager.
- Update the `OPAMDS` data source to use the SSL endpoints on your back-end RDBMS.

Note: Refer to "Enabling SSL in Other Authentication Modes" and to "SSL-Enable a Data Source" in the *Oracle Fusion Middleware Administrator's Guide* for instructions.

These actions encrypt network traffic between Oracle Privileged Account Manager and the back-end Oracle Privileged Account Manager database. They also ensure that the database is not vulnerable to exploitation by malicious elements.

Use Oracle Database Vault

Using Oracle Database Vault for production deployments of Oracle Privileged Account Manager secures the Oracle Privileged Account Manager schema and ensures that only the Oracle Privileged Account Manager application has access to the schema.

Note: Refer to "Configuring Secure Application Roles for Oracle Database Vault" in the *Oracle Database Vault Administrator's Guide* for instructions.

Thus, you prevent inadvertent or malicious access to the Oracle Privileged Account Manager application schema (and associated data) by database administrators and other implicitly authorized users. You also ensure that only the Oracle Privileged Account Manager application can access the sensitive information it maintains in the back-end Oracle Privileged Account Manager database.

2.5 Understanding Session Management Security

Currently, Oracle Privileged Account Manager only provides Session Management on SSH-enabled targets. Accessing a target through the Oracle Privileged Session Manager (Session Manager) prevents the end-user from being exposed to the password associated with the privileged account in use. Therefore, for all use-cases where the end-user does not require knowledge of the password associated with a privileged account, granting access to just the session, as opposed to the password or both the session and password, is recommended. Refer to step 3 in Section 10.3.5, "Creating a Usage Policy."

SSH Session Management support requires that the Session Manager perform the tasks associated with an SSH server. For server authentication and data privacy (such as encryption), Oracle Privileged Account Manager generates a new DSA 1024 bit (such as FIPS 186-2 compliant) SSH server key for every Oracle Privileged Account Manager instance. This server key is configurable and it can be changed by using Session Manager Configuration for key rollover.

2.6 Understanding Plug-In Security

The Oracle Privileged Account Manager plug-in framework provides for significant extensibility and customizability. However, this framework also allows custom code to interface closely with Oracle Privileged Account Manager functionality. Therefore, protecting Oracle Privileged Account Manager against malicious custom code is very important.

Because custom plug-in code can interface closely with internal Oracle Privileged Account Manager functionality, it is important that you closely scrutinize all custom plug-in code that is deployed on Oracle Privileged Account Manager. To prevent a single malicious entity from deploying malicious custom code, Oracle Privileged Account Manager enforces a requirement that two different Oracle Privileged Account Manager administrators (with different roles) are involved during the process of deploying custom plug-in code.

First, an administrator with the `OPAM_APPLICATION_CONFIGURATOR` Admin Role can add a new custom plug-in and configure it. However, the action of enabling the plug-in and determining the conditions under which it can run must be done by a second administrator with the `OPAM_SECURITY_ADMIN` Admin Role. For more information, refer to Section 13.3, "Creating a Plug-In Configuration." This design ensures that a single malicious entity cannot deploy a malicious plug-in by himself or herself.

Furthermore, because the custom plug-in code runs within the context of the Oracle Privileged Account Manager server, internal Oracle Privileged Account Manager APIs could potentially be accessible to the custom plug-in code. To pro-actively prevent this access, the Oracle Privileged Account Manager plug-in framework uses a custom class

loader that explicitly prevents access to all internal Oracle Privileged Account Manager server APIs. Therefore, the custom plug-in code must interface with internal Oracle Privileged Account Manager logic through the well-defined APIs described in Section 18.3, "Understanding the Plug-In API."

Part II

Basic Administration

This part provides information about performing basic administration tasks for Oracle Privileged Account Manager from the Console, and it contains the following chapters:

- Getting Started with Managing Oracle Privileged Account Manager
- Starting and Using the Oracle Privileged Account Manager Console
- Configuring and Managing the Servers
- Working with Targets
- Working with Service Accounts
- Configuring and Managing Agents
- Working with Privileged Accounts
- Working with Policies
- Working with Grantees
- Working with Resource Groups
- Working with Plug-Ins
- Working with Self-Service

3

Getting Started with Managing Oracle Privileged Account Manager

This chapter describes how to finish configuring Oracle Privileged Account Manager after installation.

> **Note:** You can manage Oracle Privileged Account Manager from the Console, from the command line, and by using Oracle Privileged Account Manager's RESTful interface.
>
> - For information about starting and using the Oracle Privileged Account Manager Console, refer to Chapter 4, "Starting and Using the Oracle Privileged Account Manager Console."
> - For information about starting and using the Oracle Privileged Account Manager Command Line Tool (CLI), refer to Appendix A, "Working with the Command Line Tool."
> - For information for starting and using the Oracle Privileged Account Manager RESTful interface, refer to Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface."

This chapter includes the following sections:

> **Note:** If you are using Oracle Privileged Account Manager on IBM WebSphere, refer to "Differences in Getting Started with Administering Oracle Privileged Account Manager" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for information about this topic.

3.1 Before You Begin

This chapter assumes that you have installed and configured Oracle Privileged Account Manager 11*g* Release 2 (11.1.2.3) as described in *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*.

Before starting the final configuration steps needed to start Oracle Privileged Account Manager, Oracle recommends the following:

- Read the "Configuring Oracle Privileged Account Manager"chapter in the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*.
- Review Table 3–1 to understand the default application URLs for various interfaces that you use to manage Oracle Privileged Account Manager in this release.

Table 3–1 Default Application URLs

Interface	Default URL
Oracle WebLogic Server Administrative Console	http://*adminserver_host*:*adminserver_port*/console/
Oracle Privileged Account Manager Console	http://*managedserver_host*:*managedserver_port*/oinav/opam
Oracle Privileged Account Manager Server	https://*managedserver_host*:*managedserver_sslport*/opam

- Review Table 3–2 to understand the various default ports for Oracle Privileged Account Manager in this release.

Table 3–2 Default Ports

Port Type	Default Port	Description
Oracle Privileged Account Manager Server	18102	The default SSL-enabled port for the WebLogic Managed Server on which the Oracle Privileged Account Manager server is deployed.
Oracle Privileged Account Manager Console	■ 18101 (non-SSL) ■ 18102 (SSL)	The WebLogic Managed Server port on which the Oracle Privileged Account Manager Console is available by default.
Oracle Privileged Session Manager (SSH)	1222	The default port on which Oracle Privileged Session Manager listens for SSH traffic"
WebLogic Admin Console	■ 7001 (non-SSL) ■ 7002 (SSL)	The default WebLogic Admin Server ports on which the WebLogic Admin Console is available.

- Review Table 3–3 to become familiar with the common directory variables that are used throughout this guide.

> **Note:** For additional information about these directories, and other common directories used in most Oracle Identity and Access Management installations and configurations, refer to "Identifying Installation Directories" in the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management* and "Understanding Oracle Fusion Middleware Concepts and Directory Structure" in the *Oracle Fusion Middleware Installation Planning Guide for Oracle Identity and Access Management*.

Table 3–3 Common Directories Used in Oracle Privileged Account Manager

Common Name	Description
MW_HOME	Provide the location of your Oracle Middleware Home directory. The Middleware Home contains the Oracle WebLogic Server home and one or more Oracle Home directories.
ORACLE_HOME *IAM_HOME*	Provide the location of the Oracle Home directory where the Oracle Privileged Account Manager files were installed. An Oracle home resides within the directory structure of the Middleware home.
JAVA_HOME	Provide the location used by your WebLogic server.
DOMAIN_HOME	Provide the top-level directory of the domain.
BI_DOMAIN_HOME	Provide the location of the Oracle BI Domain.

- Review the "Starting or Stopping the Oracle Stack" section in *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*, and use these instructions whenever you are instructed in this guide, to start or stop the Oracle WebLogic Administration Server (Admin Server) or any of the various Managed Servers.

3.2 Understanding ICF Connectors in Oracle Privileged Account Manager

Oracle Privileged Account Manager enables you to secure, share, audit, and manage administrator-identified account credentials. To provide these capabilities, Oracle Privileged Account Manager must be able to access and manage privileged accounts on a target system.

Connectors enable Oracle Privileged Account Manager to interact with target systems, such as LDAP or Oracle Database, and to perform Oracle Privileged Account Manager-relevant administrative operations on those systems.

Oracle Privileged Account Manager leverages connectors that are compliant with the Identity Connector Framework (ICF) standard. By using this standard, you separate Oracle Privileged Account Manager from the mechanism it uses for connecting to targets. Therefore, in addition to connectors provided by vendors such as Oracle, you are free to build, test, and deploy your own ICF connectors into Oracle Privileged Account Manager.

This section describes how Oracle Privileged Account Manager consumes these ICF connectors. It includes the following topics:

- Section 3.2.1, "About the ICF Connectors"
- Section 3.2.2, "Locating the Oracle Privileged Account Manager Connector Bundles"
- Section 3.2.3, "Consuming ICF Connectors"

Note: For more information about the Identity Connector Framework, refer to "Understanding the Identity Connector Framework" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.

3.2.1 About the ICF Connectors

Oracle Privileged Account Manager ships with the following ICF-compliant connectors that were developed by Oracle:

- Database User Management (DBUM) Connector
- Generic LDAP Connector
- Oracle Identity Manager Connector for UNIX
- SSH Connector
- SAP Connector
- Windows Local Accounts Connector

These connectors enable Oracle Privileged Account Manager to manage privileged accounts on a range of target systems belonging to the preceding types.

Oracle Privileged Account Manager can also use customer-created, ICF-compliant connectors, which empowers you to manage your proprietary systems by using Oracle Privileged Account Manager.

Note: If you are only interested in using the connectors that ship with Oracle Privileged Account Manager, then ***no further action is required*** because these connectors come pre-configured out-of-the-box.

If you want to use other Oracle connectors or a custom connector, then refer to Section 17.3, "Adding New Connectors to an Existing Oracle Privileged Account Manager Installation" for more information.

For additional information about developing ICF-compliant connectors, refer to "Developing Identity Connectors" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.

3.2.2 Locating the Oracle Privileged Account Manager Connector Bundles

Because ICF connectors are generic, and useful in numerous contexts, a given Oracle installation puts all connector bundles into a single location on the file system. All components (such as Oracle Privileged Account Manager) that rely on these connector bundles can access them from this location:

ORACLE_HOME/connectors

The connectors that are available in *ORACLE_HOME*/connectors are shipped with Oracle Identity Manager. Of all the connectors in this directory, only the five following connectors are certified to work with Oracle Privileged Account Manager for this release:

- org.identityconnectors.dbum-1.0.1116.jar
- org.identityconnectors.genericunix-1.0.0.jar
- org.identityconnectors.ldap-1.0.6380.jar
- org.identityconnectors.sap-2.0.0.jar
- org.identityconnectors.sapume-1.0.1.jar

The following connectors that are certified to work only with Oracle Privileged Account Manager are available in the ORACLE_HOME/opam-connectors location:

- org.identityconnectors.ssh-1.0.1115.jar
- WindowsLocalConnector-1.0.0.0.zip

Note: If you obtain any new ICF connectors from Oracle, you must place them in the location specified in the instructions provided.

Storing custom third-party connectors is at your discretion; however, you must ensure they can be read by Oracle Privileged Account Manager at run time.

3.2.3 Consuming ICF Connectors

Oracle Privileged Account Manager consumes ICF connectors by using the `opam-config.xml` file. The contents of this file provide the following information to Oracle Privileged Account Manager:

1. Where to pick up the ICF connector bundle (on the file system)
2. Which configuration attributes are relevant for the Oracle Privileged Account Manager use-cases
3. How to render the Oracle Privileged Account Manager Console when configuring connectivity to a target system using a particular connector

You will find the `opam-config.xml` file in the *ORACLE_HOME*`/opam/config` directory. The out-of-the-box image is configured to pick up and use the connector bundles that ship with the Oracle Identity Management Suite.

The `opam-config.xsd` file (also located in the *ORACLE_HOME*`/opam/config` directory) describes the schema for `opam-config.xml`. If you make any changes to *ORACLE_HOME*`/opam/config/opam-config.xml` file, verify them with the `opam-config.xsd` file.

Caution: Be sure to back-up the original `opam-config.xml` file before attempting to edit that file.

3.3 Starting Oracle Privileged Account Manager

This section provides high-level information about starting and working with Oracle Privileged Account Manager. This section includes the following topics:

- Section 3.3.1, "Starting WebLogic"
- Section 3.3.2, "Configuring an External Identity Store for Oracle Privileged Account Manager"
- Section 3.3.3, "Preparing the Identity Store"
- Section 3.3.4, "Assigning the Application Configurator Role to a User"

The procedures described in this section reference information and instructions contained in the following Oracle publications. If necessary, review the referenced concepts, terminology, and procedures before starting these procedures.

Table 3–4 Reference Information

For Information About	Refer to
Admin Roles	Section 2.3.1, "Administration Role Types," and Section 3.3.4, "Assigning the Application Configurator Role to a User."
System Requirements and Certification	Section 1.5, "System Requirements and Certification."
Oracle WebLogic Server concepts and terminology	*Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help* and *Oracle Fusion Middleware Securing Oracle WebLogic Server*
Creating a default authenticator in Oracle WebLogic Server	*Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help* and *Oracle Fusion Middleware Securing Oracle WebLogic Server*
Configuring an identity store in your environment	Your vendor product documentation
Configuring Oracle Virtual Directory with the LDAP-based server	"Creating LDAP Adapters" in *Oracle Fusion Middleware Administrator's Guide for Oracle Virtual Directory*
Configuring the OVD authenticator in Oracle WebLogic Server	*Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*
Connecting the Node Manager to WLST	"Node Manager Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*
Associating a Policy Store using WLST	"Setting a Node in an Oracle Internet Directory Server" and "reassociateSecurityStore" sections in the *Oracle Fusion Middleware Application Security Guide*
Associating a Policy Store using Enterprise Manager	"Reassociating with Fusion Middleware Control" in the *Oracle Fusion Middleware Application Security Guide*
Using the `idmConfigTool` command	*Oracle Fusion Middleware Integration Overview for Oracle Identity Management Suite*

Note: If you are using Oracle Privileged Account Manager on IBM WebSphere, you must start IBM WebSphere and perform some configuration steps *before* assigning the Application Configurator and invoking the Oracle Privileged Account Manager Console.

For more information about these tasks, refer to "Starting Oracle Privileged Account Manager on IBM WebSphere" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management*.

3.3.1 Starting WebLogic

Before you can start Oracle Privileged Account Manager, you must start the WebLogic servers and console.

Note:

- For detailed information about starting WebLogic and Managed Servers, refer to "Starting or Stopping the Oracle Stack" in the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*.
- You must have the appropriate Administration Role and credentials to start the server. refer to Section 2.3.1, "Administration Role Types" for more information.

1. Connect the Node Manager to WLST by running the `nmConnect` command.

   Refer to "Node Manager Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* for instructions.

2. Start the WebLogic Admin Server. For example,

   On **UNIX**, type

   ```
   MW_HOME/user_projects/domains/DOMAIN_NAME/bin/startWebLogic.sh
   ```

   On **Windows**, type

   ```
   MW_HOME\user_projects\domains\DOMAIN_NAME\bin\startWebLogic.bat
   ```

3. Start the Oracle Privileged Account Manager Managed Server.
4. Open a browser and start the WebLogic Console from the following location:

   http://*adminserver_host*:*adminserver_port*/console

3.3.2 Configuring an External Identity Store for Oracle Privileged Account Manager

This section describes how to configure a new, external identity store for Oracle Privileged Account Manager.

Note: If you are using IBM WebSphere, you must configure a *registry* rather than an external identity store. Refer to "Configuring a Registry" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for instructions.

You can configure the domain identity store using Oracle Internet Directory or Oracle Virtual Directory with a supported LDAP-based directory server. You configure the identity store in the WebLogic Server Administration Console.

Note:

- Oracle Privileged Account Manager can use any LDAP directory that is supported by Oracle WebLogic Server, as its identity store.

  For more information about configuring an identity store, refer to "Configuring the Identity Store Service" in the *Oracle Fusion Middleware Application Security Guide*.

- For information about other supported identity stores, refer to "System Requirements and Certification" in *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Navigator*.

To configure the Oracle Internet Directory authenticator in Oracle WebLogic Server:

1. Log in to Oracle WebLogic Server Administration Console, and click **Lock & Edit** in the Change Center.
2. In Oracle WebLogic Server Administration Console, select **Security Realms** from the left pane and click the realm you are configuring. For example, the default realm is *myrealm*.
3. Select the Providers tab, then select the Authentication subtab.
4. Click **New** to launch the Create a New Authentication Provider page and complete the fields as follows:
   - Name

     Enter a name for the Authentication provider. For example, **MyOIDDirectory**.
   - Type

     Select **OracleInternetDirectoryAuthenticator** from the list.

   Click **OK** to update the Authentication providers table.
5. In the Authentication providers table, click the newly added authenticator.
6. In Settings, select the Configuration tab, then select the Common tab.
7. On the Common tab, set the **Control Flag** to **SUFFICIENT**.

   Setting the Control Flag attribute for the *authenticator provider* determines the ordered execution of the Authentication providers. The following are the possible values for the Control Flag attribute:
   - **REQUIRED:** This LoginModule must succeed. Even if it fails, authentication proceeds down the list of LoginModules for the configured Authentication providers. This setting is the default.
   - **REQUISITE:** This LoginModule must succeed. If other Authentication providers are configured and this LoginModule succeeds, authentication proceeds down the list of LoginModules. Otherwise, control is returned to the application.
   - **SUFFICIENT:** This LoginModule need not succeed. If it does succeed, return control to the application. If it fails and other Authentication providers are configured, authentication proceeds down the LoginModule list.
   - **OPTIONAL:** This LoginModule can succeed or fail. However, if all Authentication providers configured in a security realm have the JAAS Control Flag set to `OPTIONAL`, the user must pass the authentication test of one of the configured providers.
8. Click **Save**.
9. Select the **Provider Specific** tab and enter the following required settings using values for your environment:
   - **Host:** Specify the host name of the Oracle Internet Directory server.
   - **Port:** Specify the port number on which the Oracle Internet Directory server is listening.
   - **Principal:** Specify the distinguished name (DN) of the Oracle Internet Directory user to be used to connect to the Oracle Internet Directory server. For example: cn=OIDUser,cn=users,dc=us,dc=mycompany,dc=com.

- **Credential:** Specify the password for the Oracle Internet Directory user entered as the Principal.
- **Group Base DN:** Specify the base distinguished name (DN) of the Oracle Internet Directory server tree that contains groups.
- **User Base DN:** Specify the base distinguished name (DN) of the Oracle Internet Directory server tree that contains users.
- **All Users Filter:** Specify the LDAP search filter that is used to show all the users below the User Base DN. Click **More Info** for details.
- **User From Name Filter:** Specify the LDAP search filter used to find the LDAP user by name. Click **More Info** for details.
- **User Name Attribute:** Specify the attribute that you want to use to authenticate, such as, cn, uid, or mail. For example, to authenticate using a user's email address you set this value to `mail`.
- **Use Retrieved User Name As Principal:** Select the check box to enable "Use Retrieved User Name As Principal."

Note: refer to Section 20.3.6, "Grantee Cannot Perform a Checkout" for additional information.

---

10. Click **Save**.
11. From the Settings for myrealm page, select the Providers tab, then select the Authentication tab.
12. Click **Reorder**.
13. Select the new authenticator and use the arrow buttons to move it into the first position in the list.
14. Click **OK**.
15. Click **DefaultAuthenticator** in the Authentication providers table to display the Settings for DefaultAuthenticator page.
16. Select the Configuration tab, then the Common tab, and select **SUFFICIENT** from the **Control Flag** list.

Note: The SUFFICIENT control flag will allow both users from an external ID store and a default authenticator to login. If it is not preferred, then this option must be switched to one of the other options. Also, if you choose SUFFICIENT, ensure that the attribute used as the user name attribute has unique values across the identity stores.

refer to Section 20.3.24, "A User is Able to Access the Grants of Another User" for more information.

---

17. In the Change Center, click **Activate Changes**.
18. Restart Oracle WebLogic Server.
19. Verify your configuration and set-up by confirming that the users present in the LDAP directory (Oracle Internet Directory or Oracle Virtual Directory) can log in to Oracle Privileged Account Manager with no issues.

To use Oracle Virtual Directory as the domain identity store, you must do the following:

- Configure Oracle Virtual Directory with an LDAP-based server as described in the "Creating LDAP Adapters" section of *Oracle Fusion Middleware Administrator's Guide for Oracle Virtual Directory*.
- Configure the OVD authenticator in Oracle WebLogic Server as described in *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.
- You must enable the "Use Retrieved User Name As Principal" option when configuring authenticators in Oracle WebLogic Server, as described in Step 9 of the preceding procedure.

> **Note:** If you are using an SSL-enabled identity store, follow the steps described in "SSL for the Identity Store Service" in the *Oracle Fusion Middleware Application Security Guide*.

3.3.3 Preparing the Identity Store

If you want to use an external LDAP server to serve as an identity store, you must seed the identity store with the necessary Oracle Privileged Account Manager users and groups.

The procedure in this section enables you to preconfigure Oracle Unified Directory (OUD) or Oracle Internet Directory (OID) for using as your LDAP Identity store.

> **Note:** The data used in the examples below is sample data. Follow the examples and replace them with appropriate data according to your LDAP server configuration.

You must complete the following steps to preconfigure the Identity Store:

1. Create a new .ldif file and name it OPAMGroups.ldif. Add the following entries to this file and save the .ldif file:

```
dn: cn=IDMSuite,dc=mycompany,dc=com
objectclass: orclContainer
objectclass: top
cn: IDMSuite

dn: cn=IDMRoles,cn=IDMSuite,dc=mycompany,dc=com
objectclass: orclContainer
objectclass: top
cn: IDMRoles

dn: cn=components,cn=IDMRoles,cn=IDMSuite,dc=mycompany,dc=com
objectclass: orclContainer
objectclass: top
cn: components

dn: cn=OPAM,cn=components,cn=IDMRoles,cn=IDMSuite,dc=mycompany,dc=com
objectclass: orclContainer
objectclass: top
cn: OPAM

dn:
cn=OPAM_APPLICATION_CONFIGURATOR,cn=OPAM,cn=components,cn=IDMRoles,cn=IDMSuite,
```

```
dc=mycompany,dc=com
objectclass: groupOfUniqueNames
objectclass: top
cn: OPAM_APPLICATION_CONFIGURATOR

dn:
cn=OPAM_USER_MANAGER,cn=OPAM,cn=components,cn=IDMRoles,cn=IDMSuite,dc=mycompany
,dc=com
objectclass: groupOfUniqueNames
objectclass: top
cn: OPAM_USER_MANAGER

dn:
cn=OPAM_SECURITY_ADMIN,cn=OPAM,cn=components,cn=IDMRoles,cn=IDMSuite,dc=mycompa
ny,dc=com
objectclass: groupOfUniqueNames
objectclass: top
cn: OPAM_SECURITY_ADMIN

dn:
cn=OPAM_SECURITY_AUDITOR,cn=OPAM,cn=components,cn=IDMRoles,cn=IDMSuite,dc=mycom
pany,dc=com
objectclass: groupOfUniqueNames
objectclass: top
cn: OPAM_SECURITY_AUDITOR
```

2. Add the OPAM Admin Role groups into Oracle Unified Directory Server using the `ldapadd` command format as described in the following example:

   ldapadd -h <OUD Server> -p <OUD port> -D <OUD Admin ID> -w <OUD Admin password> -c -f ./OPAMGroups.ldif

   The following is a sample code for this format:

   ```
   ldapadd -h localhost -p 3938 -D "cn=Directory Manager" -w "passcode1"
   -c -f ./OPAMGroups.ldif
   ```

   If you encounter an authentication error, repeat the command using `-x` with the simple bind option, as described in the following example:

   ```
   x -D "cn=Directory Manager" -w "password1" -c -f ./OPAMGroups.ldif
   ```

3.3.4 Assigning the Application Configurator Role to a User

After successful installation, there are no users with administrator roles.

In order to assign an OPAM Admin Role to a user, ensure that the user is a member of the respective OPAM LDAP groups which were created in Section 3.3.3, "Preparing the Identity Store." Assign the Application Configurator role to a user by making the user the member of the OPAM_APPLICATION_CONFIGURATOR role.

> **Note:** The Application Configurator user can have other roles in addition to this role. For more information about other Admin Roles, refer to Section 2.3.1, "Administration Role Types."

When the Application Configurator user logs in by using the following URL, that user will see a empty screen with a **Configure OPAM** link.

http://*managedserver_host*:*managedserver_port*/oinav/opam

The Application Configurator user can use this link to let the Oracle Privileged Account Manager Console know where Oracle Privileged Account Manager server is running by providing the Oracle Privileged Account Manager server's host and port.

When the Oracle Privileged Account Manager Console can successfully communicate with the Oracle Privileged Account Manager server, the Oracle Privileged Account Manager Console will be populated with content.

You are now ready to start using Oracle Privileged Account Manager.

For information about invoking and working with the Oracle Privileged Account Manager Console, refer to Chapter 4, "Starting and Using the Oracle Privileged Account Manager Console."

If you prefer using the Oracle Privileged Account Manager Command Line Tool (CLI), refer to Appendix A, "Working with the Command Line Tool."

If you prefer using the Oracle Privileged Account Manager RESTful interface, refer to Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface."

3.4 Administering Oracle Privileged Account Manager

The following table describes the basic workflows that are performed by Oracle Privileged Account Manager administrator users based on their different Admin Roles.

Note: An administrator with the *Application Configurator* Admin Role should have already configured a connection to the Oracle Privileged Account Manager servers. Refer to Section 5.2.2, "Configuring a Connection to the Oracle Privileged Account Manager Server" for more information.

Table 3–5 Administrator Workflows Based on Admin Roles

Administrator	Responsibility
Application Configurator	1. Configures and manages the Oracle Privileged Account Manager Console and servers. 2. Manages plug-in configurations.
Security Administrator	1. Evaluates Oracle Privileged Account Manager's Default Usage Policy and Default Password Policy and, if necessary, modifies these policies or creates new ones. 2. Adds targets to Oracle Privileged Account Manager. 3. Adds privileged accounts on that target. **Note**: This role cannot assign grantees to privileged accounts. 4. Assigns a Password Policy to privileged accounts. 5. Manages existing targets, accounts, and policies. 6. Manages under which conditions plug-ins can be executed. These administrators can enable or disable plug-in configurations and configure rules that control whether Oracle Privileged Account Manager executes the plug-in and in which order those rules are executed.
User Manager	1. Assigns grants to accounts. 2. Creates and manages Usage Policies as needed. 3. Assigns a Usage Policy to grants. 4. Manages existing grants and Usage Policy assignments. 5. Searches for and views plug-ins.
Security Auditor	1. Evaluates Oracle Privileged Account Manager reports.

Note: For more information about these Admin Roles, refer to Section 2.3.1, "Administration Role Types."

3.5 Working with Oracle Privileged Account Manager Self-Service

The following steps describe the basic workflow of a Self-Service user with no administrator privileges:

1. View accounts
2. Search for an account
3. Check out accounts
4. View checked-out accounts
5. Check in accounts
6. Check out a session
7. View checked out sessions
8. Check in a session
9. View an account password

Note: Refer to Chapter 14, "Working with Self-Service" for detailed information about how to perform these tasks.

4

Starting and Using the Oracle Privileged Account Manager Console

This chapter describes how to start and work with Oracle Privileged Account Manager's web user interface, known as the *Console*.

This chapter includes the following sections:

- Section 4.1, "Before You Begin"
- Section 4.2, "Invoking Oracle Privileged Account Manager's Web-Based Console"
- Section 4.3, "Navigating Oracle Privileged Account Manager's Console"

Note: You can also manage Oracle Privileged Account Manager from the command line or by using Oracle Privileged Account Manager's RESTful interface.

- Refer to Appendix A, "Working with the Command Line Tool" for information about using the command line tool.
- Refer to Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface" for information about using the RESTful interface.

4.1 Before You Begin

This chapter assumes that you have finished configuring Oracle Privileged Account Manager as described in Chapter 3, "Getting Started with Managing Oracle Privileged Account Manager."

4.2 Invoking Oracle Privileged Account Manager's Web-Based Console

You can access Oracle Privileged Account Manager's Console by opening a browser window and entering the following URL:

http://*managedserver_host*:*managedserver_port*/oinav/opam

When the Oracle Privileged Account Manager page is displayed with the Sign In screen, log in with the appropriate administrator or end user credentials.

Note: If you prefer using Oracle Privileged Account Manager's command line tool or Oracle Privileged Account Manager's RESTful interface, refer to Appendix A, "Working with the Command Line Tool" or Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface" (respectively) for detailed information about using those interfaces.

4.3 Navigating Oracle Privileged Account Manager's Console

This section provides a high-level overview of the Oracle Privileged Account Manager Console. The this section includes the following topics:

- Section 4.3.1, "Working with the Home Accordion"
- Section 4.3.2, "Working with the Administration Accordion"
- Section 4.3.3, "Working with the Reports Accordion"
- Section 4.3.4, "Working with the Configuration Accordion"
- Section 4.3.5, "Working with the Search Portlet"
- Section 4.3.6, "Working with a Search Results Table"

When you log in to Oracle Privileged Account Manager, the Console is displayed.

Access to certain features in the Console is based on your administration role (Admin Role), credentials, or delegated administrative privileges. For example, Figure 4–1 shows all of the features available in Oracle Privileged Account Manager. However, the Administration, Reports, and Configuration accordions, described later in this section, are not available to end users or to users with the *Security Administrator* role.

Figure 4–1 Oracle Privileged Account Manager Console (Full Privileges View)

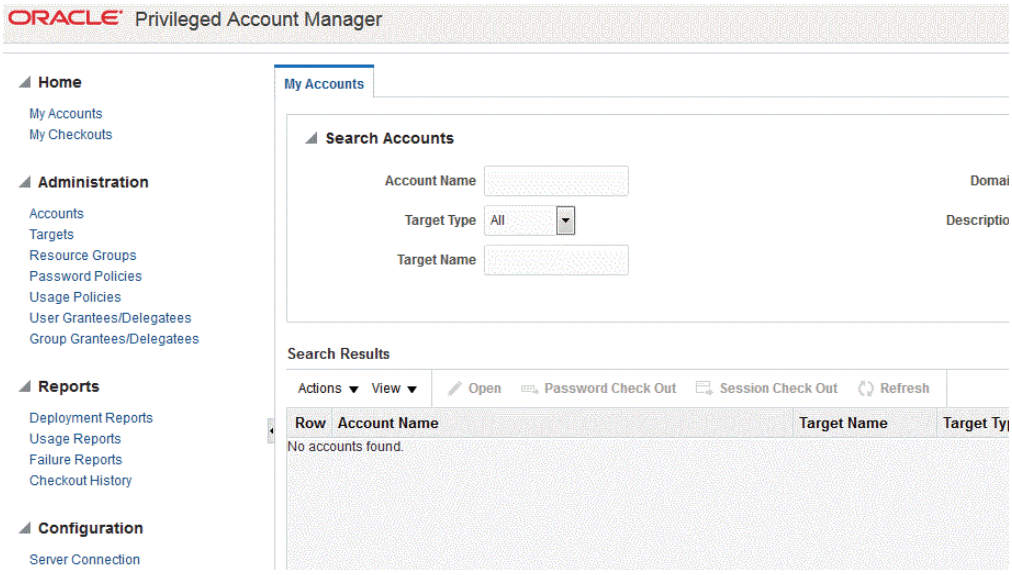
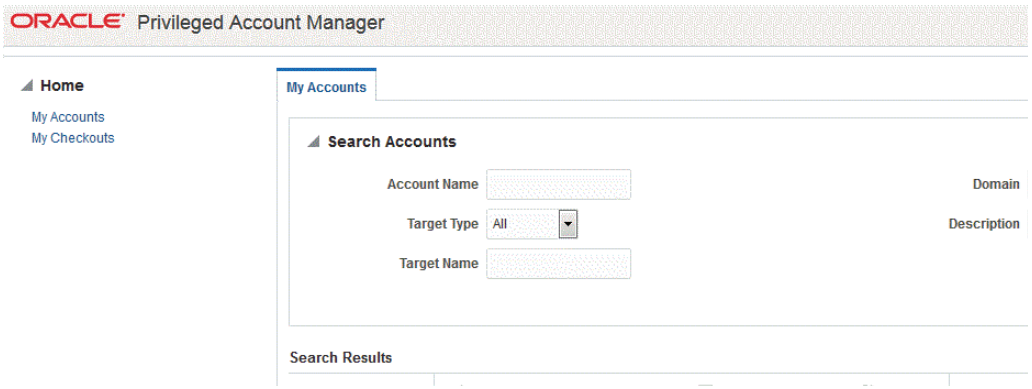


Figure 4–2 shows the Console when you log in as a Self-Service user with no administrator privileges.

Figure 4–2 Oracle Privileged Account Manager Console (Self-Service View)



Note:

- Refer to Section 2.3, "Understanding Oracle Privileged Account Manager Authorization" for more information about Admin Roles.
- Refer to Section 12.3.1, "Understanding Delegation" for more information about delegated administration privileges.

Tip: Hover your mouse over elements in the Oracle Privileged Account Manager interface (such as parameter fields or information icons) to see helpful prompts.

4.3.1 Working with the Home Accordion

The Home accordion contains the following nodes:

- **My Accounts**: Select this node to access the My Accounts page where you can search, view, open, and check out accounts where you are a grantee.
- **My Checkouts**: Select this node to access the My Checkouts page where you can view your checked out accounts, view the password for those accounts, and check in your checked out accounts.

  You must check out a privileged account to use it. Oracle Privileged Account Manager enables you to check out an account as a password or as a session. Refer to Section 9.5, "Checking Out Privileged Accounts" for more information.

Clicking either node opens a new page on the right side of the Console. Use these pages to manage your accounts.

Note:

- The My Accounts page is displayed by default when any user logs in, regardless of privileges.
- For detailed information about working with the My Accounts page or with the My Checkouts page, refer to Chapter 14, "Working with Self-Service."

4.3.2 Working with the Administration Accordion

Based on your Admin Role and credentials, the Administration accordion contains some or all of the following nodes:

- **Accounts**: Select to open the Accounts page, where you can search, open, add, and remove accounts.
- **Targets**: Select to open the Targets page, where you can search, open, add, and remove targets.
- **Resource Groups**: Select to open the Resource Groups page, where you can search, open, create, and delete resource groups.
- **Password Policies**: Select to open the Password Policies page, where you can search, open, create, and delete Password Policies.
- **Usage Policies**: Select to open the Usage Policies page, where you can search, open, create, and delete Usage Policies.
- **User Grantees / Delegatees**: Select to open the User Grantees page, where you can search, open, and view information about individual user grantees and delegatees.
- **Group Grantees** / Delegatees: Select to open the Group Grantees page, where you can search, open, and view information about a group of grantees and delegatees.

Clicking any of these nodes opens a new page on the right side of the Console. Use these pages to configure and manage Oracle Privileged Account Manager.

Note:

- For detailed information about configuring and managing Oracle Privileged Account Manager, refer to Chapter 3, "Getting Started with Managing Oracle Privileged Account Manager."
- For detailed information about configuring and managing an Oracle Privileged Account Manager server, refer to Section 5.2, "Managing an Oracle Privileged Account Manager Server."

4.3.3 Working with the Reports Accordion

Based on your Admin Role and credentials, the Reports accordion contains some or all of the following nodes:

- **Deployment Reports**: Select to open the Deployment Report page, where you can view information about how targets and privileged accounts are currently deployed in your deployment.
- **Usage Reports**: Select to open the Usage Reports page, where you can view information about how privileged accounts are being used in your deployment.
- **Failure Reports**: Select to open the Failure Reports page, where you can view information about the current state of target and account failures.
- **Checkout History**: Select to open the Checkout History page, where you can search for and review information about account checkouts.

Note: For detailed information about these reports, refer to Chapter 15, "Working with Reports."

4.3.4 Working with the Configuration Accordion

Based on your Admin Role and credentials, the Configuration accordion contains some or all of the following nodes, which represent the common global configuration properties that apply to all Oracle Privileged Account Manager servers in a cluster:

- **Server Connection**: Select to configure a connection to the Oracle Privileged Account Manager server.

  Note: Refer to Section 5.2.2, "Configuring a Connection to the Oracle Privileged Account Manager Server" for more information.

- **Server Configuration**: Select to manage the following server properties:
  - Usage Policy enforcement interval
  - Password Policy enforcement interval
  - Target connection timeout
  - Resource lock wait timeout
  - Oracle Database TDE Mode (Transparent Data Encryption)
  - User's password retrieval options
  - Identity Store search filter

  Note: Refer to Section 5.2.3, "Managing Oracle Privileged Account Manager Server Properties" for more information.

- **Plug-in Configuration**: Select to create, edit, and manage plug-in configurations for Oracle Privileged Account Manager.

  Note: Refer to Chapter 13, "Working with Plug-Ins" for more information.

- **Session Manager Configuration**: Select to configure the Session Manager properties, configure Oracle Privileged Account Manager server URLs, and SSH configuration.

  Note: Refer to Section 5.3.3, "Managing the Oracle Privileged Session Manager Properties" for more information.

- **Connector Server Configuration**: Select to create, edit, and manage Connector Server properties.

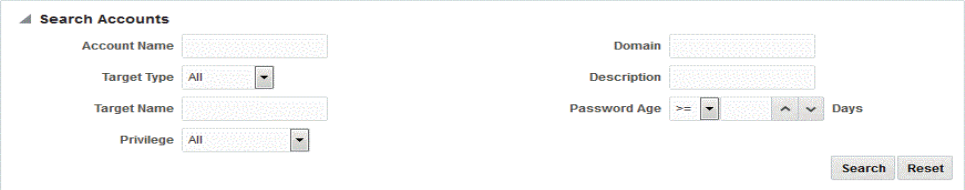
  Note: Refer to Section 5.4.3.1, "Managing a Connector Server Configuration" for more information.

4.3.5 Working with the Search Portlet

Use Oracle Privileged Account Manager's Search portlet to search for accounts, targets, policies, users, groups, and plug-ins.

You configure searches by using one or more of the parameters displayed in the portlet. The availability of different search parameters depends on the type of search you are going to perform. For example, Figure 4–3 shows the Search Accounts portlet that you use to search for privileged accounts.

Figure 4–3 Example Search Portlet



The following table describes the different search parameters and for which search types they are available:

Table 4–1 Search Portlet Parameters

Parameter Name	Description	Search Type
Account Name	Enter one or more letters of the account name.	Accounts, My Accounts, Checkout History
Target Type	Select **All** to search all target types or limit the search to only, **database**, **ldap**, **lockbox**, **sapum**, **sapume**, **ssh**, **windows**, or **unix** target types.	Accounts, My Accounts
Target Name	Enter one or more letters of the target name.	Accounts, My Accounts, Checkout History
Domain	Enter one or more letters of the domain name.	Accounts, My Accounts, Targets
Description	Enter one or more letters of the account, target, resource group, or plug-in description.	Accounts, My Accounts, Resource Groups, Plug-in Configuration
Password Age	Use the menu and **Days** field to search for accounts by password age. Choose the greater than, equal symbol (**>=**) or the less than symbol (**<**) from the menu, and enter the number of days. For example, you can use this option to search for accounts with passwords that are older than 30 days.	Accounts, Targets
Host	Enter one or more letters of the host name on which to search.	Targets
Policy Name	Enter one or more letters of the policy name.	Password Policies, Usage Policies
Policy Status	Select **All** to search all policies or limit the search to only **Active** or only **Disabled** policies.	Password Policies, Usage Policies
User Name	Enter one or more letters of the user name.	User Grantees, Checkout History
Group Name	Enter one or more letters of the group name.	Group Grantees
Start Date and End Date	Use the Calendar/Time icon to specify a date range and time in which to search.	Checkout History
Pattern	Enter one or more characters of a string in the recording of a checkout event. For example, `sync:x:5:0:sync:/sbin:/bin/sync`	Checkout History
Query Size	Use the counter to limit how many query results are returned.	Checkout History

Table 4–1 (Cont.) Search Portlet Parameters

Parameter Name	Description	Search Type
Name	Enter one or more letters of a resource group or plug-in name.	Resource Groups, Plug-in Configuration, Targets
Privilege	Select **All** to search all privileges or select the desired privilege from the drop-down list to limit the search to only the selected privilege.	Accounts, Targets, Resource Groups
Type	Select **All** to search all target types or select the desired target type from the drop-down list to select only **database**, only **ldap**, or any specific target type.	Targets
Resource Type	Select **All** to search all resource types or limit the search to only **account**, only **server**, or only **target** resource types.	Plug-in Configuration
Status	Select **All** to search all plug-in statuses or limit the search to only **Active** or only **Disabled** plug-ins.	Plug-in Configuration
Timing	Select **All** to search all plug-in timings or limit the search to only **pre** timing plug-ins or only **post** timing plug-ins.	Plug-in Configuration
Operation	Select **All** to search all plug-in operations or limit the search to only **accountpasswordchange**, **add**, **autocheckin**, **checkin**, **checkout**, **passwordcycle**, **remove**, **resetpassword**, **retrieve**, **sessioncheckout**, **showpassword**, **showpassordhistory**, **test**, or **update** operations.	Plug-in Configuration

The Search Portlet also supports the use of wildcards, as follows:

- Use the percentage symbol (**%**) to search for character strings of any length. You can also use multiple wildcards in the same search string. For example,
  - If you enter **person%**, then the results might include `person1`, `person_2`, and `person1234`.
  - If you enter **%person%**, then the results might include `dsperson`, `hrperson1`, and `hrperson2`.
- Use an underscore symbol (_) to search for a single character. You can also use multiple wildcards in the same search string. For example,
  - If you enter **person_**, then the results might include `person1`, `person2`, and `persons`.
  - If you enter **o_m_**, then the results might include `oam1`, `oem1`, `oem2`, `oem3`, and `oim1`.

The general steps for performing a search are as follows:

1. Select the appropriate node in the Home, Administration, Reports, or Configuration accordion.

   For example, to search for an account, select **Accounts**.

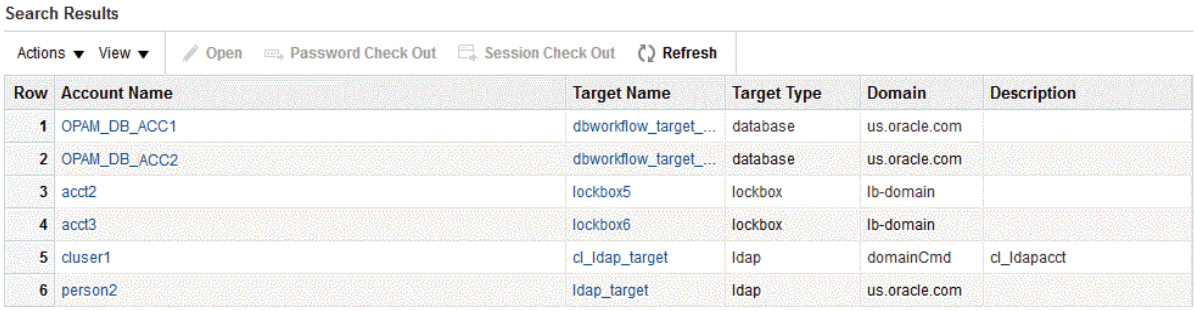
2. When the Search portlet is displayed, configure a search as follows:
   - To search for all available results, such as all accounts, do not specify any search parameters in the portlet.
   - To refine your search, use one or more of the search parameters described in Table 4–1.

     For example, to see a list of the privileged accounts on a particular LDAP target, enter one or more letters of the target's name in the **Target Name** field and select **ldap** from **Target Type** menu.

3. Click **Search**.

   The results are displayed in a Search Results table, similar to the one shown in Figure 4–4.

Figure 4–4 Example Search Results Table


Search Results

Actions ▼ View ▼ Open Password Check Out Session Check Out Refresh

Row	Account Name	Target Name	Target Type	Domain	Description
1	OPAM_DB_ACC1	dbworkflow_target_...	database	us.oracle.com	
2	OPAM_DB_ACC2	dbworkflow_target_...	database	us.oracle.com	
3	acct2	lockbox5	lockbox	lb-domain	
4	acct3	lockbox6	lockbox	lb-domain	
5	cluser1	cl_ldap_target	ldap	domainCmd	cl_ldapacct
6	person2	ldap_target	ldap	us.oracle.com	

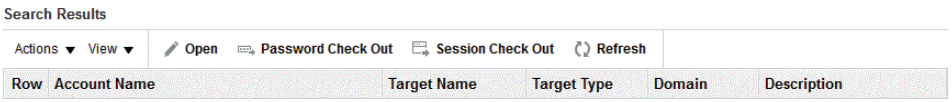
Note: You can use the **View** menu, located above the Search Results table, to manage how the search results are displayed in the table. Refer to Table 4–2 in Section 4.3.6, "Working with a Search Results Table" for more information.

4. To perform another search, click **Reset**.

4.3.6 Working with a Search Results Table

Every Search Results table has menus and icons located along the top of the table. For example, Figure 4–5 shows the menus and icons that are available after searching for targets.

Figure 4–5 Search Results Menus and Icons


Search Results

Actions ▼ View ▼ Open Password Check Out Session Check Out Refresh

Row	Account Name	Target Name	Target Type	Domain	Description

You can use these features to perform different tasks that relate to the items listed in the table.

Note: The availability of these features will change, based on what type of search was performed and on your Admin Role or administrative privileges.

- Refer to Section 2.3.1, "Administration Role Types" for information about Admin Roles.
- Refer to Section 12.3.1, "Understanding Delegation" for information about delegated administrative privileges.

Table 4–2 describes which features are available based on the type of search performed.

Table 4–2 Search Results Table Features

Feature Name	Search Type	Description
Actions	All	Click this menu and select an action to perform. **Note**: The options on this menu duplicate the task icons displayed above the table.
View	All	Click this menu and select one of the following options to control how columns are displayed in the Search Results table: ■ **Columns** > **Show All**: Displays all columns in the table. ■ **Columns** > *Column Name*: Click a column name to display or hide that column in the table. The columns are displayed (checked) by default. ■ **Columns** > **Manage Columns**: Provides a dialog box that enables you to display or hide columns. ■ **Reorder Columns**: Select this option and the Reorder Columns dialog box is displayed. Use this dialog box to select the columns and shift their order in the table.
Open	All	Click this option to open the selected account, target, policy, user grantee, group grantee, or plug-in configuration.
Password Check Out	My Accounts	Select a row in the Search Results table and click this option to check out the account's password.
Session Check Out	My Accounts	Select a row in the Search Results table and click this option to check out a session.
Refresh	My Accounts, My Checkouts, Accounts, Targets, Checkout History, Resource Groups, Plug-in Configuration, Connector Server Configuration	Click this option to re-display (refresh) the Search Results.
Check In	My Checkouts only	Click this option to check in the selected checked-out account. Refer to Section 9.6, "Checking In Privileged Accounts" for more information.
Show Password	My Checkouts, Accounts, Targets	Click this option to open the Show Current Password dialog box where you can view the current password information about a selected account or target service target. ■ **For Accounts**, this dialog lists the current Account Name and Password. ■ **For Targets**, this dialog lists the current Target Name, Service Account Name, Current Password, and Password Change Time.
Password History	Accounts, Targets	Click this option to open the Show Password History dialog box where you can view the password history for an account or a target. ■ **For Accounts**, this dialog box lists the current Account Name, Password, and Modification Time (date and time). ■ **For Targets**, this dialog box lists the Target Name, Passwords, and Modification Time (date and time).

Table 4–2 (Cont.) Search Results Table Features

Feature Name	Search Type	Description
Status	Accounts only	Click this menu and select one of the following options to limit which account results are displayed in the table: ■ **All**: Lists all accounts on the target. ■ **Checked-in Accounts**: Lists only those accounts that are currently checked-in. ■ **Checked-out Accounts**: Lists only those accounts that are currently checked-out.
Add	Accounts, Targets	Click this option to add a new account or a new target to the Oracle Privileged Account Manager repository.
Remove	Accounts, Targets	Click this option to remove the selected account or target from the Oracle Privileged Account Manager repository.
Reset Password	Accounts, Targets	Click this option to open the Reset Password dialog box where you can manually reset the password for a selected account or target service account. ■ **For Accounts**, this dialog box lists the current Account Name and Target Name. Type a password in the New Password field to create a new password for the account. ■ **For Targets**, this dialog lists the current Target Name and Service Account Name. You can either type a password in the New Password field or enable the Generate password automatically checkbox to automatically generate a new password.
Force Check In	Accounts only	Click this option to check in privileged accounts that have been checked-out by other users.
Create	Password Policies, Usage Policies, Connector Server Configuration	Click this option to create a Password Policy, Usage Policy or Connector Server Configuration. Refer Section 10.2.4, "Creating a Password Policy" for more information. Refer Section 10.3.5, "Creating a Usage Policy" for more information. Refer Section 5.4.1, "Installing and Configuring a Connector Server" for more information.
Delete	Resource Groups, Password Policies, Usage Policies, Plug-in Configuration, Connector Server Configuration	Click this option to delete a selected policy from the Oracle Privileged Account Manager repository.
Create	Resource Groups, Plug-in Configuration	Click this option to create a plug-in configuration. Refer to Section 13.3, "Creating a Plug-In Configuration" for more information.
Recording	Checkout History	Click this option to view a recording, in transcript format, of the actions taken during an account checkout.
Query By Example	Resource Groups, checkout History, Plug-in Configuration, Connector Server Configuration	Enter a value in a text field for a column. For example, to refine the search by name enter a value in the Name column.

5

Configuring and Managing the Servers

This chapter provides information that administrators must know to configure and manage an Oracle Privileged Account Manager server and an Oracle Privileged Session Manager (Session Manager) server.

This chapter includes the following sections:

Note: If you are using Oracle Privileged Account Manager on IBM WebSphere, refer to the "Differences in Configuring and Managing the Servers" section in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for information.

5.1 Understanding the Servers

This section provides a high-level overview of the following servers:

5.1.1 Oracle Privileged Account Manager Server

The Oracle Privileged Account Manager server implements the core functionality of Oracle Privileged Account Manager and makes authorization decisions that determine:

- Which targets and privileged accounts are exposed to administrators and end-users
- Which operations administrators and end-users can perform on targets, privileged accounts, and policies

In addition, the Oracle Privileged Account Manager server

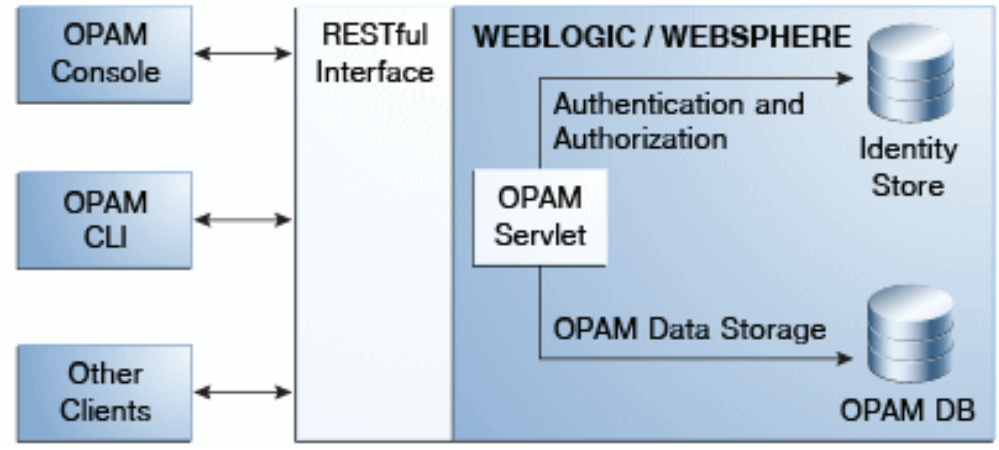
- Supports Usage and Password Policies for accounts
- Enforces its authorization decisions

- Supports authentication by using the SAML-based Oracle Security Token from OPSS Trust Services and HTTP-Basic Authentication
- Supports different Admin Roles for the Oracle Privileged Account Manager server

> **Note:** For security purposes, the Oracle Privileged Account Manager server only responds to SSL traffic.
>
> When you add the Oracle Privileged Account Manager server target to the Oracle Privileged Account Manager user interface or to the Oracle Privileged Account Manager command line tool (CLI), you must provide the SSL endpoint as https://*hostname*:*sslport*/opam.
>
> By default, WebLogic responds to SSL using port 7002 on the Admin Server and port 18102 on the Managed Server. You can use the WebLogic console to check the port for your particular instance.

The following figure illustrates the Oracle Privileged Account Manager server architecture.

Figure 5–1 Server Architecture



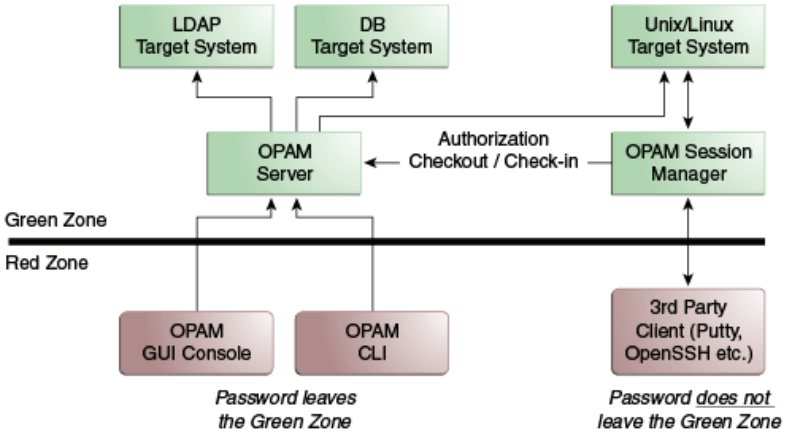
5.1.2 Oracle Privileged Session Manager Server

The Oracle Privileged Session Manager creates a single access point to target resources and enables you to manage privileged sessions to the target system through

- **Session Initiation** by
  - Providing a single control point for privileged access
  - Never exposing privileged credentials
  - Supporting any compliant, third-party clients (such as Putty, OpenSSH, etc.)
- **Session Control** by providing control through policy-based and administrator-initiated session termination and lockout.
- **Session Monitoring and Auditing** by maintaining historical records (transcripts) to support forensic analysis and audit data

The following figure illustrates how the Oracle Privileged Session Manager relates to the Oracle Privileged Account Manager server.

Figure 5–2 How Session Manager Relates to the Oracle Privileged Account Manager Server



5.1.3 Identity Connector Server

An identity connector server is required when an identity connector bundle is not directly executed within Oracle Privileged Account Manager. By using one or more identity connector servers, the ICF architecture permits Oracle Privileged Account Manager to communicate with externally deployed identity connector bundles. Identity connector servers are available for Java™ and Microsoft .NET Framework applications.

For more information refer to "Using an Identity Connector Server" of the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.

Connector servers can be configured in Oracle Privileged Account Manager, then Oracle Privileged Account Manager will be able to use the supported connectors that are deployed on the connector server.

Oracle Privileged Account Manager can continue to use the locally available connector JAR to access the target system. Additionally, you can configure more than one connector server, and associate one server with the target system. Oracle Privileged Account Manager will use this server to access the target system.

Note: For the Windows connector, you must use the .NET based identity connector server to access the target system.

5.2 Managing an Oracle Privileged Account Manager Server

This section provides information administrators need to manage an Oracle Privileged Account Manager server, which includes the following topics:

- Section 5.2.1, "Before You Begin"
- Section 5.2.2, "Configuring a Connection to the Oracle Privileged Account Manager Server"
- Section 5.2.3, "Managing Oracle Privileged Account Manager Server Properties"

5.2.1 Before You Begin

- You must be an Oracle Privileged Account Manager administrator with the *Application Configurator* Admin Role to add and manage an Oracle Privileged Account Manager server.

  Note: For more information about this Admin Role, refer to Section 2.3.1, "Administration Role Types" and Section 3.3.4, "Assigning the Application Configurator Role to a User."

- The procedures described in this chapter reference information and instructions contained in the following Oracle publications. If necessary, review the referenced concepts, terminology, and procedures before you begin configuring the Oracle Privileged Account Manager server.

Table 5–1 Reference Information

For Information About	Refer to
Admin Roles	Section 2.3.1, "Administration Role Types" and Section 3.3.4, "Assigning the Application Configurator Role to a User"
Oracle WebLogic Server concepts and terminology	*Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help* and *Oracle Fusion Middleware Securing Oracle WebLogic Server*
Adding and managing an Oracle Privileged Account Manager server on IBM WebSphere	"IBM WebSphere Identity Stores" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management*
Directory structure	"Oracle Fusion Middleware Directory Structure" in the *Oracle Fusion Middleware Installation Planning Guide*
Starting WebLogic and Managed Servers	"Starting or Stopping the Oracle Stack" in the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*

5.2.2 Configuring a Connection to the Oracle Privileged Account Manager Server

When you log into Oracle Privileged Account Manager, the Oracle Privileged Account Manager Server URL is automatically detected by default.

Use the following steps to configure a new connection to the Oracle Privileged Account Manager server from the Oracle Privileged Account Manager Console:

1. Open Oracle Privileged Account Manager by logging in to:

   http://<*opam-managedserver-host*>:<*opam-managedserver-nonsslport*>/oinav/opam

   Note: You must log in as a user with the *Application Configurator* Admin Role, or the Server Configuration page will not be accessible.

   For more information about this, and other, Admin Roles refer to Section 2.3.1, "Administration Role Types" and Section 3.3.4, "Assigning the Application Configurator Role to a User."

2. When the Oracle Privileged Account Manager Console is displayed, select **Server Connection** from the Configuration accordion.
3. When the Server Connection page is displayed, notice that the Oracle Privileged Account Manager Server URL is displayed as the **Auto-Detect URL**.

To add a different server, enter that server's **Host** name and **SSL Port** number.

> **Note:** You must provide a fully qualified host name for the **Host** value. Using *localhost* can cause problems, such as described in Section 20.3.13, "Cannot Open Session or Video Recordings."

4. Click the **Test** button to test the connection settings.

   If the server configuration tested successfully, a "`Test Succeeded`" message is displayed.

5. Click the **Apply** button to save this connection information.

5.2.3 Managing Oracle Privileged Account Manager Server Properties

You can use the Console or properties in the OPAM Global Config configuration entry to define server-level behavior for activities such as scheduler intervals, timeouts, etc. The available server properties are explained in detail in Section 5.2.3.1, "From the Console."

You can manage server properties defined in the OPAM Global Config configuration entry from two locations:

- Section 5.2.3.1, "From the Console"
- Section 5.2.3.2, "From the Command Line"

5.2.3.1 From the Console

Use the following steps to manage the Oracle Privileged Account Manager server properties from the Oracle Privileged Account Manager Console:

1. Open Oracle Privileged Account Manager by logging in to:

   http://<*opam-managedserver-host*>:<*opam-managedserver-nonsslport*>/oinav/opam

   > **Note:** You must log in as a user with the *Application Configurator* Admin Role, or the Server Configuration page will not be accessible.
   >
   > For more information about this, and other, Admin Roles refer to Section 2.3.1, "Administration Role Types" and Section 3.3.4, "Assigning the Application Configurator Role to a User."

2. When the Oracle Privileged Account Manager Console is displayed, select **Server Configuration** from the Configuration accordion.
3. When the Server Configuration page is displayed, you can modify any of the following server property options:
   - **Usage policy enforcement interval in seconds**. Specify an interval (in seconds) in which Oracle Privileged Account Manager checks accounts and then automatically checks-in the accounts that have exceeded the expiration time defined in the Usage Policy. (Default is `3600` seconds)
   - **Password policy enforcement interval in seconds**. Specify an interval (in seconds) in which Oracle Privileged Account Manager checks and then resets the password for any accounts that have exceeded the maximum password age defined in the Password Policy. (Default is `3600` seconds)

- **Target connection timeout in seconds**. Specify an interval (in seconds) in which Oracle Privileged Account Manager allows an ICF connector to wait for a response from the target system to which it is connecting.

  The default value for this setting is *20* seconds, but in some deployments where network latency is high and target systems take longer to respond, you may need to increase this value.

- **Require TDE enabled backend**. Check this box to enable Oracle Privileged Account Manager to use Transparent Data Encryption (TDE) mode. The default is TDE mode enabled.

  Enabling TDE ensures that all sensitive information stored by Oracle Privileged Account Manager (such as account passwords) is encrypted on disk.

  Unchecking the box disables TDE mode.

  Note: Oracle *strongly recommends* that you enable TDE mode for enhanced security.

  Refer to Section 2.4.7, "Hardening the Back-End Oracle Privileged Account Manager Database" for more information about using TDE mode.

- **User's password retrieval option**. Choose one of the following options from this menu to configure a password display option for users:
  - **Enable show password**. Provides only the **Show Password** button on the Checkout Account dialog. Users can click this button to view the account password, in clear text, directly on the dialog.
  - **Enable copy password**. Provides only the **Copy Password** button on the Checkout Account dialog. Users can click this button to copy the password to their clipboard for use.
  - **Enable show password and copy password**. Provides both a **Show Password** button and a **Copy Password** button on the Checkout Account dialog. Users can click the **Show Password** button to view the account password directly in the dialog box or click the **Copy Password** button to copy the password to their clipboard
- **Identity Store search filter**. Enter one of the following values to configure how Oracle Privileged Account Manager searches the Identity Store:
  - `beginswith` (Default). Oracle Privileged Account Manager searches for the results that begin with the keyword.
  - `contains`. Oracle Privileged Account Manager searches for results which contain the keyword.
- **Resource lock wait timeout in seconds**: Specify an interval (in seconds) which defines the maximum time allowed for an operation to obtain a transaction lock on a resource. (Default is *120* seconds.)

4. When you are finished, click the **Apply** button to save these configuration settings.

5.2.3.2 From the Command Line

To access the OPAM Global Config configuration entry and modify these server properties, use the `getconfig` and the `modifyconfig` commands from the command line.

Note: Refer to Section A.2.1, "`getconfig` Command" and Section A.2.3, "`modifyconfig` Command" for detailed information about using these commands.

Refer to Section 17.2, "Securing Data On Disk" for more information about enabling or disabling TDE mode from the command line.

5.3 Managing the Oracle Privileged Session Manager Server

This section provides information administrators need to manage a Session Manager Server, which includes the following topics:

- Section 5.3.1, "Before You Begin"
- Section 5.3.2, "Configuring a Connection to the Oracle Privileged Session Manager Server"
- Section 5.3.3, "Managing the Oracle Privileged Session Manager Properties"

5.3.1 Before You Begin

- You must be an administrator with the *Application Configurator* Admin Role or the *Security Administrator* role to view the Session Manager Configuration page.
- Only administrators with the *Application Configurator* Admin Role can modify any of the settings on the Session Manager Configuration page.

Note: For more information about these Admin Roles refer to Section 2.3.1, "Administration Role Types" and Section 3.3.4, "Assigning the Application Configurator Role to a User."

5.3.2 Configuring a Connection to the Oracle Privileged Session Manager Server

Use the following steps to configure the Oracle Privileged Session Manager server from the Oracle Privileged Account Manager Console:

1. Open Oracle Privileged Account Manager by logging in to:

   https://<*opam-managedserver-host*>:<*opam-managedserver-sslport*>/opam

2. When the Oracle Privileged Account Manager Console is displayed, select **Session Manager Configuration** from the Configuration accordion.

   Use the properties on the Session Manager Configuration page to configure the Session Manager. Refer to Section 5.3.3, "Managing the Oracle Privileged Session Manager Properties" for instructions.

Note: You cannot run two instances of Oracle Privileged Session Manager on the same machine.

5.3.3 Managing the Oracle Privileged Session Manager Properties

Use the following steps to manage the Session Manager properties from the Oracle Privileged Account Manager Console:

Note:

- You can also configure Session Manager properties by using the Oracle Privileged Account Manager RESTful interface. Refer to Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface" for more information.
- You cannot use the Oracle Privileged Account Manager Command Line Tool (CLI) to configure Session Manager properties.

1. Open Oracle Privileged Account Manager and navigate to the Session Manager Configuration page as described in Section 5.3.2, "Configuring a Connection to the Oracle Privileged Session Manager Server."
2. When the Session Manager Configuration page is displayed, configure the following options:
   - **Session Monitoring Update Interval in seconds**. Specify an interval (in seconds) in which Session Manager checks all checked-out sessions and updates their transcripts. Session Manager automatically terminates any sessions that have exceeded the expiration time defined in the Usage Policy. (Default is *60* seconds).
   - **Max recording size per Session**. Specify the maximum recording size that is allowed per session (in KB). This recording size limits how much data a user can use after checking out a session. When this quota is reached, the session is automatically terminated. The default is values is 10240.
   - **Oracle Privileged Account Manager URLs**. Use this table to manage an array of Oracle Privileged Account Manager servers to which Session Manager can connect:

     Note: Notice that the Oracle Privileged Account Manager Server URL is displayed by default in the first row of the table, as the **Auto-Detect URL**.

     Clicking the **Add** button removes the Auto-Detect URL. After adding one or more rows to the table, you must click **Remove** and remove all rows to use the Auto-Detect URL instead. The Auto-Detect URL is only displayed when the table is empty.

     The Oracle Privileged Account Manager Server URL is multi-valued to allow for High Availability (HA).

     Session Manager maintains the server list and, when required, uses it on a round-robin basis for connections to Oracle Privileged Account Manager. Connection attempts are made against all configured servers until one succeeds or all configured URLs are exhausted.

     – To add one or more Oracle Privileged Account Manager Server URLs, click **Add**.

       When the new row is displayed in the table, enter the URL of an Oracle Privileged Account Manager server into the blank field. For example,

       `https://<opamserver_host>:<port>/opam`

  - To delete one or more Oracle Privileged Account Manager Server URLs from the table, select the row and click **Remove**.

- **SSH Configuration**. Use the following options to configure the connection details to be displayed for session checkouts:
  - **Listener Port**: Provide the reserved SSH port on which the Session Manager listener protocol is listening. The value must be greater than 1024 and it defaults to 1222.
  - **Session Checkout Instructions**: Enter an instruction message to be displayed when users check out a session. This message should describe the information a user must provide to connect to the Session Manager server by using a regular SSH client.

    For example:

    ```
    ssh -p <port> <opamuser>:<targetname>:<accountname>@<sessionmgrhost>
    Use opam password on password prompt
    ```

Note: **Windows Agent Count** is the number of windows agents deployed. This is a read only field.

---

3. When you are finished, click the **Apply** button to save these configuration settings.
4. Restart Oracle Privileged Account Manager.

Note: Any time you change the Session Manager configuration, you must restart Oracle Privileged Account Manager for the changes to take effect.

For the detailed instructions you need to check out and check in sessions, refer to Section 14.7, "Checking Out Privileged Account Sessions."

5.4 Managing a Connector Server

This section discusses the following topics:

- Section 5.4.1, "Installing and Configuring a Connector Server"
- Section 5.4.2, "Installing the Windows Connector"
- Section 5.4.3, "Managing a Connector Server Configuration in Oracle Privileged Account Manager"

5.4.1 Installing and Configuring a Connector Server

The following is an overview of the procedure to configure the Connector Server:

- Section 5.4.1.1, "Installing and Configuring The Connector Server"
- Section 5.4.1.2, "Configuring SSL Between the Connector Server and the Windows Target"
- Section 5.4.1.3, "Configuring SSL Between Oracle Privileged Account Manager and the .NET Connector Server"
- Section 5.4.1.4, "Enabling Logging"

5.4.1.1 Installing and Configuring The Connector Server

Identity connector servers are available for Java™ and Microsoft .NET Framework applications. See one of the following references depending on the requirements in your environment:

- For Java based connector server, refer to the "Using the Java Connector Server" section of the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.
- For .NET based connector server refer to the "Using the Microsoft .NET Framework Connector Server" section of the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.

The Windows Local connector is implemented in .NET. You must therefore deploy this connector to a .NET framework-based Connector Server.

Before you deploy the .NET Connector Server, check if you have installed Microsoft .NET Framework 4.5 on the same computer where you are installing the Connector Server.

If you are using .NET version 4.0 or above, perform the following steps in the .NET Connector Server configuration file(ConnectorServer.exe.config):

1. Search for startup.
2. Replace content between <startup> and </startup> with the following:

   Note: While configuring the Connector Server on a Windows platform, the version for the supportedRunTime parameter must be listed from the highest version to the lowest version.

   In other words, the supported runtime version must be listed in the descending order.

```
<startup>
   <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5"/>
   <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>
   <supportedRuntime version="v2.0.50727"/>
</startup>
```

3. Specify a port for the connector server in the connector configuration file. The port is required for the Windows connector to connect to the Windows Connector server:

```
<add key="connectorserver.port" value="8759" />
```

4. In the command line, set a custom key for the .NET Connector Server. You must use the /setkey command-line argument, as described in the following procedure:
   a. Navigate to the directory where the .NET Connector Server was installed. The default directory is `C:\Program Files\Identity Connectors\Connector Server`.
   b. Run the following command:

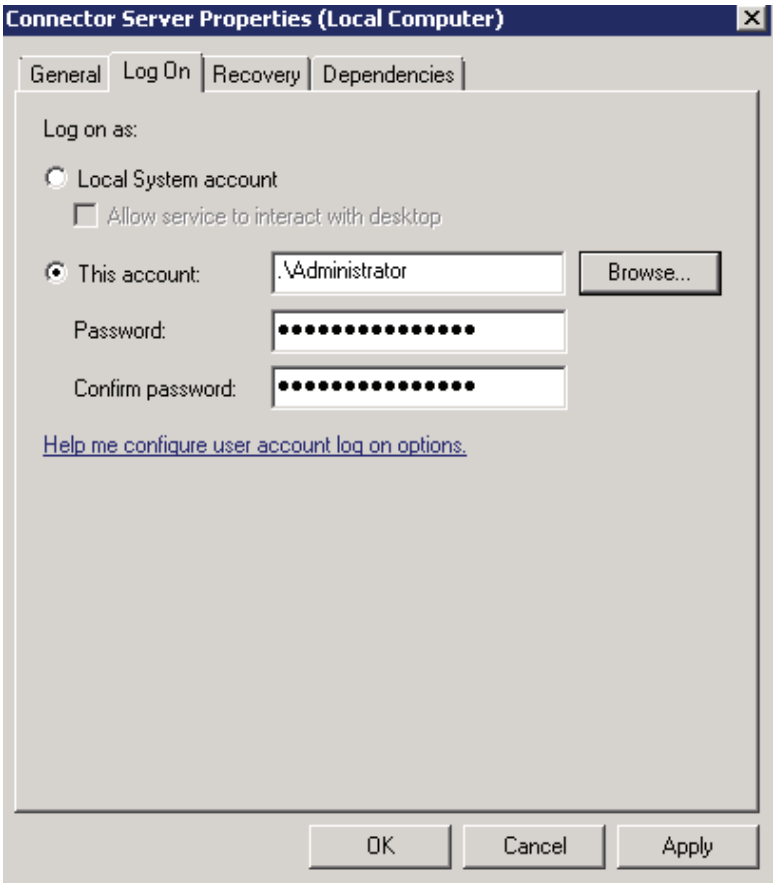
      ```
      ConnectorServer.exe /setkey NEW_KEY
      ```

      In this command, NEW_KEY is the value for the new key. This key is required by Oracle Privileged Account Manager to connect to the .NET Connector Server.
5. Save the file.

6. When managing the domain accounts with non-admin (delegated user) then you must start the Connector Server with Admin User Privileges.

   Figure 5–3 shows how to start the Connector Server with Admin User Privileges.

Figure 5–3 How to Start the Connector Server with Admin User Privileges



Note: To run the Connector Server service from the console or directly from command line, it must be run as an administrative user.

You can start this by performing one of the following procedures:

- From the Microsoft Services Console:
  a. Right click the **Connector Server,** and click **Properties.**
  b. Navigate to the Log on tab and select the **This Account** option for log on. Provide the credentials of a user who is a member of the Administrators group. Then, start the connector server.
- From the Command Line: Login to the connector server machine with the credentials of a user who is a member of Administrators group. Open the console and navigate to the directory where the .NET Connector Server was installed. From this directory, run the following command:

```
ConnectorServer.exe /run
```

5.4.1.2 Configuring SSL Between the Connector Server and the Windows Target

The connection from the connector server to the target Windows machine is always secure. The Windows connector uses the "ADS_SECURE_AUTHENTICATION" option.

Refer to the "ADS_SECURE_AUTHENTICATION" section in the following link for more information:

```
http://msdn.microsoft.com/en-us/library/aa772247%28v=vs.85%29.aspx
```

Note: Using "Secure Socket Layer (SSL)" is not supported as this option is not allowed by the WinNT API used by the connector. However, as mentioned in this section, the connection from the connector server to the Windows target is still secure.

5.4.1.3 Configuring SSL Between Oracle Privileged Account Manager and the .NET Connector Server

To configure the SSL between Oracle Privileged Account Manager and the .NET connector server, you must perform the following procedure:

1. On the Windows machine hosting the connector server, export the certificate. To do so:
   a. Click **Start** and then **Run.**
   b. Enter the following command, and then click **OK:**

      ```
      mmc
      ```

      The Microsoft Management Console is displayed.
   c. From the File menu, select **Add/Remove Snap-in.**
   d. In the Add or Remove Snap-ins dialog box, select **Certificates** from the available snap-ins list, and then click **Add.**

e. In the Certificates snap-in dialog box, select **Computer account,** and then click **Next.**

f. In the Select Computer dialog box, select **Local computer,** and then click **Finish.**

g. In the Add or Remove Snap-ins dialog box, click **OK.**

h. In the left pane of the Console Root window, expand **Certificates (Local Computer), Personal,** and then select **Certificates.** All requested certificates will be displayed in the right pane.

i. Right-click the certificate, select **All Tasks,** and then click **Export.** The Certificate Export Wizard is displayed.

j. On the Welcome to the Certificate Export Wizard page, click **Next.**

k. On the Export Private Key page, select the **No, do not export the private key** option, and then click **Next.**

l. On the Export File Format page, select **Base-64 encoded X.509(.CER)** and click **Next.**

m. On the File to Export page, in the **File name** field, enter the name and location to which the certificate must be exported and then click **Next.** The following is a sample location:

```
C:\ WindowsLocalCer
```

n. On the Completing the Certificate Export Wizard page, click **Finish.**

A dialog box with a message that the export was successful is displayed.

o. Click **OK.**

2. Configure the connector server for SSL. To do so:

a. Create a certificate store and add the certificate created in Step 1 of this procedure. To add a certificate store:

In a command window, enter the following:

```
C:\>certutil -f -addstore sslstore C:\WindowsLocalCer.cer
```

This command creates a new certificate store with the name 'sslstore' and adds the certificate WindowsLocalCer.cer to this store.

Note:

- Ensure that the certificate store with the name mentioned in the preceding command does not already exist. In other words, the certificate store mentioned in the ConnectorServer.exe.Config file must have only one certificate. If there are more than one certificates, then the Connector Server will not start.

  Run the following command to view the number of certificates present in the certificate store:

  ```
  C:\>certutil -viewstore STORE_NAM
  ```

- If the certificate has been exported with a private key (for example, .pfx file), then you must import it into the certificate store named 'sslstore' using the MMC console.

---

b. Navigate to the location where Connector Server is installed and locate the ConnectorServer.exe.Config file.

c. In a text editor, open the ConnectorServer.exe.Config file for editing and change the values of the following lines:

From:

```
<add key="connectorserver.usessl" value="false" />
<add key="connectorserver.certificatestorename"
value="ConnectorServerSSLCertificate" />
```

To:

```
<add key="connectorserver.usessl" value="true" />
<add key="connectorserver.certificatestorename" value="sslstore" />
```

d. Restart the Connector Server.

3. Configure Oracle Privileged Account Manager for SSL.

Import the target system certificate into the JDK used by Oracle Privileged Account Manager. To do so:

a. Copy the certificate generated in Step 1 of this procedure to the computer on which Oracle Privileged Account Manager is deployed.

b. Import the target system certificate into the JDK used by Oracle Privileged Account Manager (running on Oracle WebLogic Application Server) by running the following command:

`keytool -import -keystore` *`MY_CACERTS`* `-file` *`CERT_FILE_NAME`* `-storepass` *`PASSWORD`*

In the above command, *MY_CACERTS* is the full path and name of the certificate store. *CERT_FILE_NAME* is the full path and name of the certificate file, and *PASSWORD* is the password of the keystore.

The following is a sample command:

```
keytool -import -keystore
/home/OPAM/jrockit_160_14_R27.6.5-32/jre/lib/security/cacerts -file
/home/WindowsLocalCer.cer -storepass changeit
```

Note: For more information on adding an SSL certificate to the Oracle Privileged Account Manager certificate store, refer to Section 17.1, "Configuring Oracle Privileged Account Manager to Communicate With Target Systems Over SSL."

5.4.1.4 Enabling Logging

For information about enabling logging in the .NET connector server, refer to the "Configuring Trace Settings" section of *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager.*

The Windows connector uses the built-in logging mechanism of the .NET framework. The log level is set in the .NET Connector Server configuration file (ConnectorServer.exe.config).

To enable logging for the Windows Local connector:

1. Go to the directory where the ConnectorServer.exe.config file is installed. The default directory is C:\Program Files\Identity Connectors\Connector Server. The ConnectorServer.exe.config file must be present in this directory.
2. In the ConnectorServer.exe.config file, add the lines shown in bold text:

```
<system.diagnostics>
  <trace autoflush="true" indentsize="4">
    <listeners>
      <remove name="Default" />
      <add name="myListener" type="System.Diagnostics.TextWriterTraceListener"
initializeData="c:\connectorserver2.log" traceOutputOptions="DateTime">
        <filter type="System.Diagnostics.EventTypeFilter"
initializeData="Information" />
      </add>
    </listeners>
  </trace>
  <switches>
    <add name="WindowsLocalSwitch" value="4" />
  </switches>
</system.diagnostics>
```

   The `value="4"` sets the log level to Verbose. This value can be set as follows:

Table 5–2 Log Levels

Value	Log Level
value="4" or value="Verbose"	Verbose level. Most granular.
value="3" or value="Information"	Information level.
value="2" or value="Warning"	Warning level.
value="1" or value="Error"	Error level.
value="0"	No logging.

   However, remember that the logging level has a direct effect on the performance of the .NET Connector Server.

3. You must stop and then restart the .NET Connector Server service.

5.4.2 Installing the Windows Connector

Oracle Privileged Account Manager can support the management of Windows local accounts using the Windows connector based on Identity Connector Framework (ICF).

The Windows connector is based on the Microsoft .NET Framework. It requires to be deployed on a .NET connector server.

5.4.2.1 Installing the Connector in the Connector Server

The following is the procedure to configure the Windows Connector. To install the connector in the connector server:

1. Stop the Connector Server.
2. Copy the "WindowsLocal.Connector-1.0.0.0.zip" file from the following location to the Windows target machine:

   `$ORACLE_HOME\opam-connectors\msft_windowslocal\bundle`

3. Unzip the .zip file and copy all the contents to `C:\Program Files\Identity Connectors\Connector Server`.
4. Restart the Connector Server.

5.4.3 Managing a Connector Server Configuration in Oracle Privileged Account Manager

The connector server is configured by specifying values for the properties mentioned in Table 5–3.

Table 5–3 Connector Server Configuration Properties

Property	Description
Name	Name for the configuration
Description	Description of the configuration
Host	The connector server host
Port	The connector server port
Server Key	The key that is used to connect to the connector server
SSL	SSL enabled/disabled
Timeout in Seconds	Timeout in seconds for a connector server operation

Refer to "Working with Oracle Privileged Account Manager's RESTful Interface" for more information about configuring connector servers through the RESTful Interface.

5.4.3.1 Managing a Connector Server Configuration

You can perform the following actions to manage the connector server configuration from the console:

- Section 5.4.3.1.1, "Creating a Connector Server Configuration"
- Section 5.4.3.1.2, "Opening a Connector Server Configuration"
- Section 5.4.3.1.3, "Deleting a Connector Server Configuration"

5.4.3.1.1 Creating a Connector Server Configuration

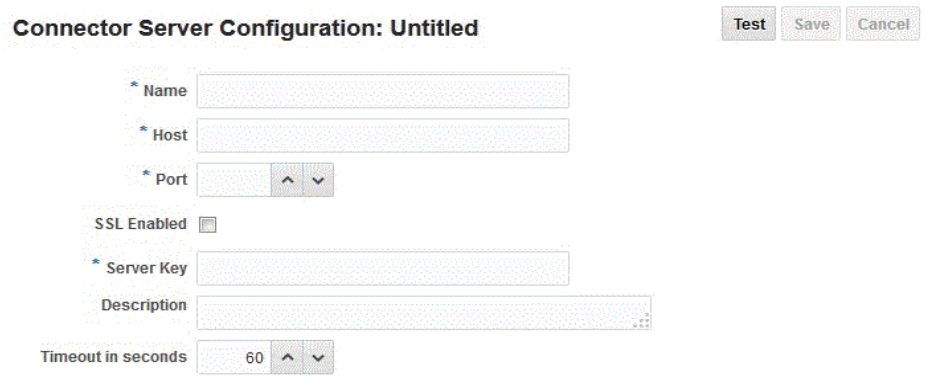
You can create a connector server configuration from the console by specifying values in the Connector Server Configuration page, as described in the following procedure:

> **Note:** Only administrators with the *Application Configurator* Admin Role can create connector server configurations.

1. Select **Connector Server Configuration** in the Configuration accordion.
2. When the Connector Server Configuration page is displayed, click **Create** in the Search Results toolbar.
3. A new "Connector Server Configuration: Untitled" page is displayed. You can configure the Connector Server by specifying values for the properties displayed in this page.

   Figure 5–4 shows the untitled connector server configuration page.

Figure 5–4 Example of the Connector Server Configuration Page



See Also: Table 5–3, " Connector Server Configuration Properties" for detailed information about the configuration properties for a connector server

4. After specifying the values, click **Test** to check if the configuration is successful. If a test fails, the "Test Failed" message is displayed it the dialog box. In this case, reconfigure the values and retry. If the test is successful, the "Test succeeded" message is displayed it the dialog box.
5. To save a successful connector server configuration, click **Save.**

5.4.3.1.2 Opening a Connector Server Configuration

You can search for and open a connector server configuration from the console as described in the following procedure:

1. Select **Connector Server Configuration** in the Configuration accordion.
2. When the Connector Server Configuration page is displayed, use the Search portlet parameters to configure your search.

   For example:

   - To search for the "TestConnectorServer" configuration, you can type t, test, or testconnector into the **Name** field.
   - To search for all existing connector server configurations, do not specify any search parameters.
3. Click **Search.** The results will be populated in the Search Results table.

5.4.3.1.3 Deleting a Connector Server Configuration

You can delete a connector server configuration from the console as described in the following procedure:

Note: Only administrators with the *Application Configurator* Admin Role can delete connector server configurations.

1. Select **Connector Server Configuration** in the Configuration accordion.

2. When the Connector Server Configuration page is displayed, use the Search portlet to locate the configuration you want to delete.
3. Select the "Row number" of the connector server configuration number from the Search Results table and then click **Delete.**

6

Working with Targets

This chapter describes the different tasks you can perform when working with targets in Oracle Privileged Account Manager.

This chapter includes the following sections:

Note:

- You can also use Oracle Privileged Account Manager's command line tool or Oracle Privileged Account Manager's RESTful interface to perform many of the tasks described in this chapter.

  If you prefer using these interfaces instead of the Oracle Privileged Account Manager Console, refer to Appendix A, "Working with the Command Line Tool" or Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface" for instructions.

- You must be an Oracle Privileged Account Manager administrator with the *Security Administrator* Admin Role to add, edit, or remove targets.

6.1 What Are Targets?

A target is a software system that contains, uses, and relies on user, system, or application accounts.

You cannot create targets in, or delete targets from, your environment by using Oracle Privileged Account Manager. Rather, Oracle Privileged Account Manager manages existing targets that were provisioned using other mechanisms.

When you "add" a target in Oracle Privileged Account Manager, you are creating a reference to that target. In effect, you are registering the target and asking Oracle Privileged Account Manager to manage it. When you "remove" a target from Oracle Privileged Account Manager, you are only removing that reference.

Oracle Privileged Account Manager supports database, LDAP, lockbox, SAP UM, SAP UME, SSH, UNIX, and Windows target types.

A *lockbox* target provides password vault-like functionality in Oracle Privileged Account Manager. That is, it provides a secure mechanism for storing the passwords (or any kind of sensitive information) associated with privileged accounts in your deployment. This target type is different from the other conventional Oracle Privileged Account Manager target types in the following ways:

- Oracle Privileged Account Manager does not interact with lockbox target systems. There is no connectivity to, or operations performed against, these systems.
- Oracle Privileged Account Manager does not manage the password lifecycle or reset passwords associated with accounts on lockbox targets.
- Password modifications are handled out-of-band and updated into Oracle Privileged Account Manager as an administrative action. Therefore, Oracle Privileged Account Manager does not randomize the passwords; but rather, they stored as given by the administrator.

A lockbox target may be preferable when you want to centrally store and securely grant privileged account passwords without having Oracle Privileged Account Manager automatically manage those accounts on the target systems. For example, if you want to control how and when the passwords on the those target systems are modified, as opposed to allowing Oracle Privileged Account Manager do so.

Additionally, a lockbox target may be useful when an appropriate ICF connector is unavailable for a specific target type, but you still want to manage access to that system through Oracle Privileged Account Manager.

6.2 Adding and Configuring Targets in Oracle Privileged Account Manager

This section discusses the following topics:

6.2.1 Adding a Target

Note: When adding a target of any Target Type (except lockbox), you must configure a service account (also called an *unattended* account) with privileges that enable that account to

- Search for accounts on the target system
- Modify the passwords of accounts on the target system

You must never use the same account as a service account *and* as a privileged account to be managed by Oracle Privileged Account Manager.

For additional information about service accounts, see the description for attended and unattended accounts in Section 1.2.1, "Features" and refer to Chapter 7, "Working with Service Accounts."

> **Note:** If you are using Oracle Privileged Account Manager on IBM WebSphere, refer to the "Differences When Adding Targets to Oracle Privileged Account Manager on IBM WebSphere" section in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for information about this topic.

Perform the following steps to add a target that Oracle Privileged Account Manager can manage:

1. Log in to Oracle Privileged Account Manager.
2. Select **Targets** from the Administration accordion to open the Targets page.
3. Click **Add**, which is located on the Search Results table toolbar, to open a new Target. A new "Untitled" page is opened, which will contain the three following tabs:

   > **Note:** Only the General tab is active at this point. The Privileged Accounts and Member-of tabs do not become active until you create and save the target.

   - **General**

     This tab generally contains three areas which are used to specify their respective parameters for the target. The three areas are:

     - Basic Configuration
     - Advanced Configuration
     - Custom Attributes

   - **Privileged Accounts**

     This tab lists the privileged accounts currently being managed on the target and enables you to add, open, and remove the accounts that are managed by that target.

   - **Member-of**

     This tab contains a table listing the different resource groups, which the privileged account is a member of.

4. On the General tab, select the **Target Type** drop-down list to select a target type (**database**, **ldap**, **lockbox, sapum, sapume, ssh, unix,** or **windows**), and then set the remaining configuration parameters and custom attributes.

   > **Note:** When you set the target type, the new "Untitled" page refreshes and the parameters change based on your selection.
   >
   > - Refer to Section 6.2.2, "Configuring a Target" for information about configuration parameters for each target type.
   > - Refer to Section 6.2.3, "Configuring Custom Attributes for a Target" for information about configuring custom attributes for a target.
   > - Refer to Section 6.2.4, "Copying Third-Party JARs" for information about copying third-party jars for a target.

5. After setting the target configuration parameters, click **Test** to check the configuration of the target.

   If the configuration is valid, a "`Test Succeeded`" message is displayed.

6. Click **Save** to add your new target on the Oracle Privileged Account Manager server.

   Oracle Privileged Account Manager automatically assigns a Target GUID and you can view this read-only value at the bottom of the Basic Configuration parameters section.

You can now associate this target with a privileged account. For detailed instructions, refer to Section 9.2, "Adding Privileged Accounts into Oracle Privileged Account Manager."

6.2.2 Configuring a Target

The following sections describe the available parameters for each target:

- Section 6.2.2.1, "Configuring the Database Target"
- Section 6.2.2.2, "Configuring the LDAP Target Type"
- Section 6.2.2.3, "Configuring the Lockbox Target Type"
- Section 6.2.2.4, "Configuring the UNIX Target Type"
- Section 6.2.2.5, "Configuring the Windows Target Type"
- Section 6.2.2.6, "Configuring the SSH Target Type"
- Section 6.2.2.7, "Configuring the SAP UM Target Type"
- Section 6.2.2.8, "Configuring the SAP UME Target Type"

Note: You must specify all of the required attributes indicated by an asterisk (*****) symbol.

6.2.2.1 Configuring the Database Target

When you select the "database" target type, the following regions are displayed:

- Basic Configuration:

  This region contains the basic configuration parameters for which the values can be specified while creating a database target type. Refer to Table 6–1 for the description of these parameters.

- Advanced Configuration:

  This region contains the optional advanced configuration parameters for which values can be specified while creating a database target type. Refer to Table 6–2 for the description of these parameters.

Table 6–1 Basic Configuration Parameters for the Database Target Type

Parameter Name	Description
Name	Enter a name for the new target.
Description	Enter a description for this target.
Organization	Enter the name of an organization to associate with the target.

Table 6–1 (Cont.) Basic Configuration Parameters for the Database Target Type

Parameter Name	Description
Domain	Enter the domain of the target server.
Password Policy	Select a Password Policy to apply to the target's service account. Oracle Privileged Account Manager uses this policy to auto-generate passwords.
Connector Server	Select a connector server from the drop-down list to specify the connector server to be used. Default is `None`.
Enable Password Rollover	Enable this box to allow Oracle Privileged Account Manager to automatically change (rollover) the service account password for this target to a randomized value according to the **Expire password after** setting that is specified in the assigned Password Policy. **Note**: Password rollover for target service accounts is similar to password expiration for privileged accounts. If a password has not been changed by the expiration date configured in the associated Password Policy, then Oracle Privileged Account Manager will automatically change the password to a randomized value.
Host	Enter the host name of the target server.
Database Connection URL	Enter the JDBC URL used to identify the target system location. For example, `Oracle:jdbc:oracle:thin:@<host>:<port>:<sid>` **Note**: Oracle Privileged Account Manager supports the Oracle, MSSQL, Sybase, and MySQL database types. Refer to the *Oracle Identity Manager Connector Guide for Database User Management* for information about special options that are supported.
Admin User Name (Service Account)	Enter the administrator' name to use when connecting to this target. **Note**: If you are using the `sys` user name, you must enter `internal_logon=sysdba` in the **Connection Properties** field, which is located in the Advanced Configuration area. This entry is not required for "system."
Admin User Password (Service Account Password)	Enter the user's password.
Database Type	Select the type of database (**Oracle**, **MSSQL**, **Sybase**, or **MySQL**) for which the connector will be used. If you select an Oracle database target, then no driver jar is required. For other target systems, you must copy third-party jars. Refer to Section 6.2.4.1, "Copying Third-Party JARs for the Database Target" for more information.
Member-of Resource Group	Search for and select a resource group with which this target can be associated.

The following table discusses the optional advanced configuration parameter:

Table 6–2 Advanced Configuration Parameters for the Database Target Type

Parameter Name	Description
Connection Properties	Enter connection properties to use while configuring a secured connection. These properties must be name-value pairs given in following format: `prop1=val1#prop2=val2`

6.2.2.2 Configuring the LDAP Target Type

When you select the "ldap" target type, the following regions are displayed:

- Basic Configuration

  This region contains the basic configuration parameters for which values can be specified while creating an ldap target type. Refer to Table 6–3 for the description of these parameters.

- Advanced Configuration

  This region contains the optional advanced configuration parameters for which values can be specified while creating an ldap target type. Refer to Table 6–4 for the description of these parameters.

Table 6–3 Basic Configuration Parameters for the LDAP Target Type

Parameter Name	Description
Name	Enter a name for the new target.
Description	Enter a description for this target.
Organization	Enter the name of an organization to associate with the target.
Domain	Enter the domain of the target server.
Password Policy	Select a Password Policy to apply to the target's service account. Oracle Privileged Account Manager uses this policy to auto-generate passwords.
Connector Server	Select a connector server from the drop-down list to specify the a connector server to be used. Default is `None`.
Host	Enter the host name of the target server.
TCP Port	Enter the TCP/IP port to use when communicating with the LDAP server. You can use the up/down arrow icons to increment this value.
SSL	Enable this box to use Secure Socket Layer (SSL) when connecting to the LDAP server. **Note**: For SSL connectivity, you must import an SSL certificate to the J2EE container hosting Oracle Privileged Account Manager. For more information, refer to Section 17.1, "Configuring Oracle Privileged Account Manager to Communicate With Target Systems Over SSL."
Principal (Service Account)	Enter the distinguished name (DN) to use when authenticating to the LDAP server. For example, cn=admin
Password (Service Account Password)	Enter the user's password.
Base Contexts	Enter one or more starting points in the LDAP tree to use when searching the tree for users on the LDAP server or when looking for groups where the user is a member. Use a pipe (`|`) to separate values.
Account User Name Attribute	Enter the attribute to be used as the account's user name. (Default is *`uid`*).
Member-of Resource Group	Search for and select a resource group with which this target can be associated.

The following table discusses the optional advanced configuration parameters:

Table 6–4 Advanced Configuration Parameters for the LDAP Target Type

Parameter Name	Description
Uid Attribute	Enter the name of the LDAP attribute that is mapped to the Uid attribute.
LDAP Filter for Retrieving Accounts	Enter an LDAP filter to control which accounts are returned from the LDAP resource. If you do not specify a filter, Oracle Privileged Account Manager returns only those accounts that include all of the specified object classes.
Password Attribute	Enter the name of the LDAP attribute that holds the password. When changing a user's password, Oracle Privileged Account Manager sets the new password to this attribute
Account Object Classes	Enter one or more object classes to use when creating new user objects in the LDAP tree. Type each object class on its own line. Do not use commas or semicolons to separate entries. Some object classes require you to specify them in their class hierarchy, using a pipe (`|`) to separate the values.

6.2.2.3 Configuring the Lockbox Target Type

When you select the "lockbox" target type, the Basic Configuration region is displayed. This region contains the basic configuration parameters for which values can be specified while creating a lockbox target type. Refer to Table 6–5 for the description of these parameters.

Table 6–5 Basic Configuration Parameters for the Lockbox Target Type

Parameter Name	Description
Name	Enter a name for the new target.
Description	Enter a description for this target.
Organization	Enter the name of an organization to associate with the target.
Domain	Enter the domain of the target server.
Host	Enter the host name of the target server.
Member-of Resource Group	Search for and select a resource group with which this target can be associated.

Note: You can add configuration parameters to this list by editing the `opam-config.xml` file as described in Section 3.2.3, "Consuming ICF Connectors."

6.2.2.4 Configuring the UNIX Target Type

When you select the "unix" target type, the following regions are displayed:

- Basic Configuration

  This region contains the basic configuration parameters for which values can be specified while creating a unix target type. Refer to Table 6–6 for the description of these parameters.

- Advanced Configuration

This region contains the optional advanced configuration parameters for which values can be specified while creating a unix target type. Refer to Table 6–7 for the description of these parameters.

Table 6–6 Basic Configuration Parameters for the Unix Target Type

Parameter Name	Description
Name	Enter a name for the new target.
Description	Enter a description for this target.
Organization	Enter the name of an organization to associate with the target.
Domain	Enter the domain of the target server.
Password Policy	Select a Password Policy to apply to the target's service account. Oracle Privileged Account Manager uses this policy to auto-generate passwords.
Connector Server	Select a connector server from the drop-down list to specify a connector server to be used. Default is `None`.
Enable Password Rollover	Enable this box to allow Oracle Privileged Account Manager to automatically change (rollover) the service account password for this target to a randomized value according to the **Expire password after** setting that is specified in the assigned Password Policy. **Note**: Password rollover for target service accounts is similar to password expiration for privileged accounts. If a password has not been changed by the expiration date configured in the associated Password Policy, then Oracle Privileged Account Manager will automatically change the password to a randomized value.
Host	Enter the host name of the target server.
Port	Enter the port (Default port is *22*) used to connect with the UNIX server. You can use the up/down arrow icons to increment this value. **Note**: Only the SSH protocol is supported.
Login User (Service Account)	Enter the user name to use when connecting to this target.
Login User Password (Service Account Password)	Enter the user's password.
Login Shell Prompt	Enter the shell prompt to display when you log in to the target. For example, **$** or **#**. **Note**: When using sudo authorization, the prompts for the login user and the sudo root account may be different. For example, jdoe's shell prompt might be **$**, but that prompt may change to **#** after a sudo to root. In such cases, you must specify both symbols within square brackets **[ ]**. The default value, `[$#%>~]`, consists of all the commonly used UNIX shell prompts and will work for most situations.
Sudo authorization	Enable this box if the user requires sudo authorization. *Do not* enable this box for the root user. **Note:** When using sudo authorization, the UNIX connector requires that certain conditions must be met in the target system, such as a specific configuration in the sudoers file. For information about these conditions, refer to "Creating a Target System SUDO User Account for Connector Operations" in the *Oracle Identity Manager Connector Guide for UNIX*.
Member-of Resource Group	Search for and select a resource group with which this target can be associated.

The following table discusses the optional advanced configuration parameters:

Table 6–7 Advanced Configuration Parameters for the Unix Target Type

Parameter Name	Description
Command timeout	Specify how long (in milliseconds) to wait for the command to complete before terminating that command.
Password Expect Expressions	Specify the expressions displayed on the target when setting the user's password. For example, if the `Enter password` and `Re-enter password` expressions are displayed when you run the `passwd` command, then the value for this field can be `enter password,re-enter password`. **Note**: You can provide a regular expression here. Use a comma to separate the two expressions.
Pre-password expectExpression	When you run the `passwd` command on some targets, prompts can be displayed before the password prompts appear. Specify the prompt expression and the expected input value, using a comma to separate these values.

sudo password expectExpression	Specify the password prompt to be displayed when running a command in sudo mode. (Default value is `password`)
	Note: This is the prompt you will receive when you type sudo -v after sudo has been setup, if you are using a sudo user.
	If you see a different prompt such as the following where oracle is the user:
	`[sudo] password for Oracle:`
	Then it is a dynamic prompt and you must change it to default static prompt so that the connector can expect the prompt to enter the password.
	The default password prompt can be setup using a sudo user by adding the following command in the "Defaults" section:
	`Defaults passprompt="password:"`

6.2.2.5 Configuring the Windows Target Type

When you select the "windows" target type, the Basic Configuration region is displayed. This region contains the basic configuration parameters for which values can be specified while creating a windows target type. Refer to Table 6–8 for the description of these parameters.

Table 6–8 Basic Configuration Parameters for the Windows Target Type

Parameter Name	Description
Name	Enter a name for the new target.
Description	Enter a description for this target.
Organization	Enter the name of an organization to associate with the target.
Domain	Enter the domain of the target server.
Password Policy	Select a Password Policy to apply to the target's service account. Oracle Privileged Account Manager uses this policy to auto-generate passwords.
Connector Server	Select a connector server from the drop-down list to specify a connector server to be used. Default is `None`. **Note:** A Windows target requires a connector server with a Windows Connector installed on it. Refer to Section 5.4, "Managing a Connector Server" for more information about configuring a connector server.

Table 6–8 (Cont.) Basic Configuration Parameters for the Windows Target Type

Parameter Name	Description
Enable Password Rollover	Enable this box to allow Oracle Privileged Account Manager to automatically change (rollover) the service account password for this target to a randomized value according to the "Expire password after" setting that is specified in the assigned Password Policy. **Note**: Password rollover for target service accounts is similar to password expiration for privileged accounts. If a password has not been changed by the expiration date configured in the associated Password Policy, then Oracle Privileged Account Manager will automatically change the password to a randomized value.
Host	Enter the host name of the target server.
Administrators Account	Enter the user name of the Administrator account. **Note:** The format for AdminName can be any of the following: ▪ `MachineName\Username` ▪ `DomainName\Username` You can give IP Address of the machine as Hostname if the AdminName is given in the format DomainName\Username.
Administrators Password	Enter the password of the Administrator account.
Member-of Resource Group	Search for and select a resource group with which this target can be associated.

6.2.2.6 Configuring the SSH Target Type

When you select the "ssh" target type, the Basic Configuration region is displayed. This region contains the basic configuration parameters for which values can be specified while creating an ssh target type. Refer to Table 6–9 for the description of these parameters.

Note: Some examples of network devices that support SSH are routers, firewalls, and hypervisors. Refer to Appendix C, "Working with the SSH Connector" for detailed information on how to add customizations to work with your specific network device.

The customization process involves creating scripts and framing regular expressions. Refer to the following sections for detailed information about these steps:

- Section C.2, "Creating Scripts"
- Section C.3, "Framing the Search Regex"

Table 6–9 Basic Configuration Parameters for the SSH Target Type

Parameter Name	Description
Name	Enter a name for the new target.
Description	Enter a description for this target.
Organization	Enter the name of an organization to associate with the target.
Domain	Enter the domain of the target server.
Password Policy	Select a Password Policy to apply to the target's service account. Oracle Privileged Account Manager uses this policy to auto-generate passwords.

Table 6–9 (Cont.) Basic Configuration Parameters for the SSH Target Type

Parameter Name	Description
Connector Server	Select a connector server from the drop-down list to specify a connector server to be used. Default is `None`.
Enable Password Rollover	Enable this box to allow Oracle Privileged Account Manager to automatically change (rollover) the service account password for this target to a randomized value according to the **Expire password after** setting that is specified in the assigned Password Policy. **Note**: Password rollover for target service accounts is similar to password expiration for privileged accounts. If a password has not been changed by the expiration date configured in the associated Password Policy, then Oracle Privileged Account Manager will automatically change the password to a randomized value.
Host	Enter the host name of the target server.
Manage Privilege Mode Password	Enable this box to allow Oracle Privileged Account Manager to manage the Privilege Mode Password of this target for Cisco devices. **Note:** The privilege Mode Password is used by some Cisco devices to enter privilege mode where privileged commands can be executed. If this option is selected, an account called "PRIVILEGE_MODE_ACCOUNT" will be created under the target in Oracle Privileged Account Manager. Security Administrators can use this account to manage the privilege mode password of that Cisco device. For example, when the password of this account is reset, the privilege mode password of the Cisco device will also be reset. When the password is reset on this account, the script defined for UPDATE_ACCOUNT operation will be used to reset the privilege mode password on the Cisco device. Refer to the Section C.2, "Creating Scripts" and Section C.4.1.1, "Contents Of the Script Files" for detailed information about scripts.
Port	Enter the port (default port is *22*) used to connect with the SSH server. You can use the up/down arrow icons to increment this value. **Note**: Only the SSH protocol is supported.
Login User Name (Service Account)	Enter the user name to use when connecting to this target.
Password (Service Account Password)	Enter the password of the user that is used to connect to this target.
Properties File Path	Enter the full path of the .properties file.
Search Regex	Enter the regex (regular expression) that must be used to fetch users, roles, or both from the user search output obtained from the target.
Login Shell Prompt	Enter the shell prompt to display when you log in to the target. For example, **$** or **#**.
Privilege Mode Password	This field is optional. Enter the password of the privilege mode, to access the privilege mode. Specify a value for this parameter only if you are using Cisco, else, you can ignore this field.
Member-of Resource Group	Search for and select a resource group with which this target can be associated.

6.2.2.7 Configuring the SAP UM Target Type

When you select the "sapum" target type, the following regions are displayed:

- Basic Configuration

This region contains the basic configuration parameters for which values can be specified while creating an sapum target type. Refer to Table 6–10 for the description of these parameters.

- Advanced Configuration

  This region contains the optional advanced configuration parameters for which values can be specified while creating an sapum target type. Refer to Table 6–11 for the description of these parameters.

> **Note:** You must copy third-party jars for this target. Refer to Section 6.2.4.2, "Copying Third-Party JARs for the SAPUM and SAPUME Targets" for more information.

Table 6–10 Basic Configuration Parameters for the SAPUM Target Type

Parameter Name	Description
Name	Enter a name for the new target.
Description	Enter a description for this target.
Organization	Enter the name of an organization to associate with the target.
Domain	Enter the domain of the target server.
Password Policy	Select a Password Policy to apply to the target's service account. Oracle Privileged Account Manager uses this policy to auto-generate passwords.
Connector Server	Select a connector server from the drop-down list to specify a connector server to be used. Default is `None`.
Enable Password Rollover	Enable this box to allow Oracle Privileged Account Manager to automatically change (rollover) the service account password for this target to a randomized value according to the **Expire password after** setting that is specified in the assigned Password Policy. **Note**: Password rollover for target service accounts is similar to password expiration for privileged accounts. If a password has not been changed by the expiration date configured in the associated Password Policy, then Oracle Privileged Account Manager will automatically change the password to a randomized value.
Host	Enter the host name of the target server.
User	Enter the name of the service account.
Password	Enter the password of the service account.
SAP System Number	Enter the system number of the SAP target. The default value is `00`.
Client	Enter name of the SAP client setting. The default value is `000`.
SAP Destination Name	Enter a unique resource name that defines the destination which must be created.
Master System	Enter the RFC destination value that is used to identify the SAP system.
Dummy Password	Enter the dummy password for the connector to use during a Create User provisioning operation
Member-of Resource Group	Search for and select a resource group with which this target can be associated.

The following table discusses the optional advanced configuration parameters:

Table 6–11 Advanced Configuration Parameters for the SAPUM Target Type

Parameter Name	Description
CUA Mode	Password propagation from master to child systems
Password Propagate to Child System	Password propagation from master to child systems

6.2.2.8 Configuring the SAP UME Target Type

When you select the "sapume" target type, the following regions are displayed:

- Basic Configuration

  This region contains the basic configuration parameters for which values can be specified while creating an sapume target type. Refer to Table 6–12 for the description of these parameters.

- Advanced Configuration

  This region contains the optional advanced configuration parameters for which values can be specified while creating an sapume target type. Refer to Table 6–13 for the description of these parameters.

> **Note:** You must copy third-party jars for this target. Refer to Section 6.2.4.2, "Copying Third-Party JARs for the SAPUM and SAPUME Targets" for more information.

Table 6–12 Basic Configuration Parameters for the SAPUME Target Type

Parameter Name	Description
Name	Enter a name for the new target.
Description	Enter a description for this target.
Organization	Enter the name of an organization to associate with the target.
Domain	Enter the domain of the target server.
Password Policy	Select a Password Policy to apply to the target's service account. Oracle Privileged Account Manager uses this policy to auto-generate passwords.
Connector Server	Select a connector server from the drop-down list to specify a connector server to be used. Default is `None`.
Enable Password Rollover	Enable this box to allow Oracle Privileged Account Manager to automatically change (rollover) the service account password for this target to a randomized value according to the "Expire password after" setting that is specified in the assigned Password Policy. **Note**: Password rollover for target service accounts is similar to password expiration for privileged accounts. If a password has not been changed by the expiration date configured in the associated Password Policy, then Oracle Privileged Account Manager will automatically change the password to a randomized value.
Host	Enter the host name of the target server.
UME URL	Enter the URL of the SPML service.

Table 6–12 (Cont.) Basic Configuration Parameters for the SAPUME Target Type

Parameter Name	Description
User Id	Enter the name of the service account.
Password	Enter the password of the service account.
Dummy Password	Enter the dummy password for the connector to use during a Create User provisioning operation
Member-of Resource Group	Search for and select a resource group with which this target can be associated.

The following table discusses the optional advanced configuration parameters:

Table 6–13 Advanced Configuration Parameters for the SAPUME Target Type

Parameter Name	Description
Logon Name Initial Substring	Enter a set of characters to support full reconciliation for the English language. For other languages, enter all characters of that language. Sample value: abcdefghijklmnopqrstuvwxyz1234567890
Log SPML Request	Enter "yes" to print the SPML request. The default value is `no`.

6.2.3 Configuring Custom Attributes for a Target

Custom attributes are optional parameters that can be used to store custom attributes and values. You can use these parameters to store additional information about the target. For example, you can define the data center name for a Unix target, define the Oracle Home path for a Oracle database target, and so on. You can use these attributes to provide more information about target systems to administrators. The custom attributes can also be used to pass such additional information to plug-ins.

You can configure a custom attribute by adding a new row and specifying values for the Attribute Name and Attribute Value columns. For multivalued attributes, you must add another row with the same Attribute Name and specify the next value in the Attribute Value column.

You can configure custom attributes for any target type by adding a new row and specifying a value in the "Attribute Name" column, and clicking **Save.**

6.2.4 Copying Third-Party JARs

This section discusses the procedure to copy third-party jars for the Database, SAPUM, and SAPUME targets. Depending on the target that you are configuring, perform one of the following procedures:

- Section 6.2.4.1, "Copying Third-Party JARs for the Database Target"
- Section 6.2.4.2, "Copying Third-Party JARs for the SAPUM and SAPUME Targets"

6.2.4.1 Copying Third-Party JARs for the Database Target

If you select an Oracle database target, then no driver jar is required. For other target systems, you must copy one of the following third-party jars:

- **For MSSQL**: Copy `sqljdbc4.jar`.

- **For MySQL**: Copy `mysql-connector-java-5.1.20-bin.jar`.
- **For Sybase**: Copy `jconn4.jar`.

You can use one of the following options to copy the jars:

Option 1: Copy the third-party jars to the WebLogic domain `/lib` directory, as described in the "Adding JARs to the Domain /lib Directory" section in *Oracle Fusion Middleware Developing Applications for Oracle WebLogic Server*.

Option 2: Modify the connector jars to include the third-party jars as follows:

1. Make a back-up copy of the DBUM connector bundle, which is available in the following location:

   ORACLE_HOME`/connectors/dbum/bundle/`
   `org.identityconnectors.dbum-1.0.1116.jar`

2. Create a `temporary/lib` folder and place the third-party jars in this folder.
3. Update the bundle with the third-party jar as shown below:

   ```
   jar -uvf org.identityconnectors.dbum-1.0.1116.jar lib/JAR_NAME
   ```

4. Delete the `temporary/lib` folder.
5. Restart all Oracle Privileged Account Manager processes for all changes to take effect.

For more information, refer to the "Installing the Connector on the Connector Server" section of the *Oracle Identity Manager Connector Guide for Database User Management*.

6.2.4.2 Copying Third-Party JARs for the SAPUM and SAPUME Targets

> **Note:** Ensure that you are using version 3.0.2 or later of the sapjco3.jar file. To download files from the SAP Web site, you must have access to the SAP service marketplace with Software Download authorization.

To download and copy the third-party jars and external code files to the required locations:

1. Download the SAP Java connector file from the SAP Web site as follows:
   a. Open the SAP Java Connector page by selecting **Application Platform, Connectivity, Connectors, SAP Java Connector,** and **Tools & Services.**
   b. On the SAP Java Connector page, links for files that you can download are displayed on the right pane. Click the link for the SAP JCo release that you want to download.
   c. In the dialog box that is displayed, specify the location in which you want to save the file.
2. From the saved location, extract the contents of the file that you download.
3. Copy these third-party jars to the WebLogic domain `/lib` directory, as described in the "Adding JARs to the Domain /lib Directory" section of *Oracle Fusion Middleware Developing Applications for Oracle WebLogic Server*.

4. Copy the RFC files into the required directory on the Oracle Identity Manager host computer, and then modify the appropriate environment variable so that it includes the path to this directory:

   - On Microsoft Windows:

     Copy the sapjco3.dll file into the *winnt\system32* directory. Alternatively, you can copy these files into any directory and then add the path to the directory in the "PATH" environment variable.

   - On Solaris and Linux:

     Copy the libsapjco3.so file into the */usr/local/jco* directory, and then add the path to this directory in the LD_LIBRARY_PATH environment variable.

5. On a Microsoft Windows platform, ensure that the msvcr80.dll and msvcp80.dll files are in the c:\WINDOWS\system32 directory. If required, both files can be downloaded from various sources on the Internet.

6. If you are using IBM WebSphere Application Server, perform the following steps:

   a. Copy the following files to WEBSPHERE_HOME/AppServer/lib:

      – libsapjco3.so

      – sapidoc3.jar

      – sapjco3.jar

      For example, copy the preceding files to the `/home/shareuser/R2PS1ST1WAS/IBM/WebSphere/AppServer/lib` location.

   b. Update the *PROFILE_HOME*/bin/setupCmdLine.sh file as shown in the following example:

      ```
      WAS_CLASSPATH="$WAS_HOME"/properties:"$WAS_HOME"/lib/startup.jar:"$
      WAS_HOME"/lib/bootstrap.jar:"$WAS_HOME"/lib/lmproxy.jar:"$WAS_HOME"
      /lib/urlprotocols.jar:"$WAS_HOME"/lib/sapjco3.jar:"$WAS_HOME"/lib/s
      apidoc3.jar:"$JAVA_HOME"/lib/tools.jar
      ```

7. Restart the server for the changes in the environment variable to take effect.

8. To check if SAP JCo is correctly installed in a command window, run one of the following commands:

   ```
   java -jar JCO_DIRECTORY/sapjco3.jar
   java -classpath JCO_DIRECTORY/sapjco3.jar com.sap.conn.jco.rt.About
   ```

   In the preceding commands, *JCO_DIRECTORY* is the location where the sapjco3.jar file was copied.

   Figure 6–1 shows the dialog box that is displayed. The JCo classes and JCo library paths must be displayed in this dialog box.

Figure 6–1 Dialog Box Displayed on Running the SAP JCo Test

This is a screenshot of the dialog box that is displayed when you run the SAP JCo test. The JCo classes and the JCo library paths displayed in this dialog box indicate that SAP JCo is correctly installed.

****

6.3 Searching for Targets

If you have administrator privileges, you can search for targets using the following criteria or a combination of these items:

- Name
- Type (**All**, **database**, **ldap**, **lockbox, sapum, sapume, ssh, unix,** or **windows**)
- Host
- Domain
- Description
- Password Age

- Privilege

To search for a target, perform the following procedure:

1. Select **Targets** in the Administration accordion.
2. When the Targets tab is displayed, use the Search portlet parameters to configure your search. For example,
   - To search for all LDAP targets, select **ldap** from the **Type** menu.
   - To search for all available targets, do not specify any search parameters.
3. Click **Search**.

   Review your search results in the Search Results table.

6.4 Opening a Target

You can open a target to review and edit the target's configuration parameters and its associated privileged account parameters.

Use one of the following methods to open a target:

- Click **Name** (an active link) in the Search Results table.
- Select the target's Row number, then click **Actions** and select the **Open** option from the drop-down list.

The Target: *TargetName* page opens where you can access the target and privileged account information.

6.5 Managing a Target's Service Account Password

Oracle Privileged Account Manager provides several options for managing a target's service account passwords, including:

- Showing passwords
- Viewing password history
- Resetting passwords
- Enabling password rollover

Administrators with the *Security Administrator* Admin Role can perform these password management tasks by using the Oracle Privileged Account Manager Console, command line tool, or REST API.

Note:

- For information about managing passwords by using the Console, refer to Section 7.3, "Managing Service Account Passwords."
- For command line instructions, refer to Section A.5, "Working with Targets."
- For REST API instructions, refer to Section B.6, "Target Resource."

Oracle Privileged Account Manager audits password management actions to keep a track of password access.

Note: The procedures for showing and resetting a privileged account password are different from the procedures described in this section. Refer to Section 9.8, "Managing Privileged Account Passwords" for information.

6.6 Removing Targets from Oracle Privileged Account Manager

To remove a target, select the target from the Search Results table and then click the **Remove** icon.

WARNING: When you remove a target, you also remove all information about the target that is stored in Oracle Privileged Account Manager (including privileged accounts).

Before removing a target, it is critical that you first capture all relevant information from that target. For example, save the target's service account password and any current passwords that are associated with the privileged accounts on the target.

7

Working with Service Accounts

This chapter provides background information about OPAM service accounts, including an example for creating those accounts.

The topics in this chapter include:

- Section 7.1, "Understanding Service Accounts"
- Section 7.2, "Creating Service Accounts"
- Section 7.3, "Managing Service Account Passwords"

7.1 Understanding Service Accounts

Before adding a target to Oracle Privileged Account Manager, you must configure an *OPAM service account* (also called an *unattended* account) for that target. OPAM service accounts (service accounts) enable Oracle Privileged Account Manager to connect to and manage target systems.

You use an OPAM service account to configure the credentials for a target system.

Note:

- Service accounts do not apply for lockbox-type targets.
- You must never use the same account as a service account *and* a privileged account to be managed by Oracle Privileged Account Manager.

A service account must have sufficient privileges to perform all Oracle Privileged Account Manager-related operations on the target system, such as:

- Searching for and viewing details about the accounts in the target, which is used for all operations such as looking up and adding privileged accounts on the system to Oracle Privileged Account Manager, locating the account during checkout, etc.
- Changing account passwords in the target, which is used for operations involving password changes such as checkout, check-in, resetpassword, etc.
- Changing self password, which is used for resetting target service account passwords and changing the password of the service account itself.

7.2 Creating Service Accounts

This section provides information about creating a service account to use when connecting to a target system.

> **Note:** Never use the same account as both a service account *and* a privileged account to be managed by Oracle Privileged Account Manager.

The methods for creating a service account and assigning privileges to that account depend on the target system. For example, the steps for creating accounts and assigning roles on an Oracle Database system are different from the steps for a UNIX operating system.

The following examples illustrate two methods for creating a service account:

> **Note:** These examples are only provided as a reference. You can achieve the same result by using other means.

On an Oracle Database System:

1. Use SQLPLUS and connect as the `sys user`.
2. Run the following commands to create the `opamsrvc` account:

```
connect sys/<password> as sysdba
create user opamsrvc identified by <password>;
grant connect, alter user, select on dba_users to opamsrvc
```

On a Linux System:

1. Use Linux and connect as `root`.
2. Run the following commands to create the `opam_service` account:

```
$ useradd -d /home/opam_service -m -g root -G bin,daemon,sys,adm,disk,wheel
-o -u 0 opam_service
$ passwd opam_service
```

7.2.1 Creating a Target System SUDO User Account for Connector Operations

Oracle Privileged Account Manager uses a target system account for performing reconciliation and provisioning operations. On all supported target systems, this account must be either the root user or sudo user.

To create a target system user account with the minimum permissions required to perform connector operations, perform the following procedure:

1. If SUDO is not installed on the target system, then install it from the installation media.
2. Use the `visudo` command to edit and customize the /etc/sudo file according to your requirements.

Note: If you cannot use the `visudo` command to edit the `sudoers` file, then:

1. Enter the following command:

   ```
   chmod 777 /etc/sudoers
   ```

2. Make the required changes in the `sudoers` file.
3. Enter the following command:

   ```
   chmod 440 /etc/sudoers
   ```

For example, if you have a group named `mqm` on the Linux server and require all members of the group to act as SUDO users with all possible privileges, then the `sudoers` file must contain the following line:

```
mqm ALL= (ALL) ALL
```

This example is only a sample configuration. If you need other group members or individual users to be SUDO users with specific privileges, then edit this file as was done for the sample value `mqm`.

Therefore, the SUDO user must have the privileges required to run these commands.

3. Edit the same `sudoers` file so that the SUDO user stays validated for 10 minutes after being validated once. You may need to increase the timeout if the reconciliation operation takes longer than 10 minutes and if you encounter errors such as "Permission denied". At the beginning of each operation, the connector validates the user using `sudo -v` option so that the operation stays validated for a maximum of 10 minutes. After carrying out the operation, the connector runs the `sudo -k` to kill the validation.

   Add the following line under the `# Defaults specification` header:

   ```
   Defaults timestamp_timeout=10
   ```

   This is a prerequisite for this connector to work successfully.

4. Edit the same `sudoers` file so that every time a command is run in SUDO Admin mode, the SUDO user is prompted for the password. Add the following line under the `# Defaults specification` header:

   ```
   Defaults timestamp_timeout=10
   ```

   This is a prerequisite for this connector to work successfully.

5. Create a SUDO user as follows:

   a. Enter the following command:

      useradd -g *group_name* -d /home/*directory_name* -m *user_name*

      In this command:

      - *group_name* is the SUDO users group for which there is an entry in the `/etc/sudoers` file.

      - *directory_name* is the name of the directory in which you want to create the default directory for the user.

   b. In the .bash_profile file, which is created in the /home/*directory_name* directory, add the following lines to set the `PATH` environment variable:

```
PATH=/usr/sbin:$PATH
export PATH
```

6. In the sudo user's .bashrc, .cshrc, or .kshrc file, which is created in the sudo user's home directory, add the following line to change the prompt end character from $ (dollar sign) to # (pound sign):

   ```
   PS1="[\\u@\\h:\\w]#"
   ```

   The encrypted passwords in the shadow file contain $ (dollar sign), which matches the default prompt end character. You must change the prompt end character to ensure that changes made to the shadow file are reconciled correctly.

7. Login with the sudo user.
8. Run the `sudo -k` command on the target system to clear the validation.
9. Run the `sudo -v` command on the target system and ensure that the password prompt is displayed.

   The connector would not work if the sudo user is not prompted for password at this step.

7.3 Managing Service Account Passwords

Oracle Privileged Account Manager provides the following options for managing a target's service account passwords:

- Section 7.3.1, "Showing Service Account Passwords"
- Section 7.3.2, "Viewing the Password History"
- Section 7.3.3, "Resetting Service Account Passwords"
- Section 7.3.4, "Understanding Service Account Password Rollover"

Administrators with the *Security Administrator* Admin Role can perform these password management tasks by using the Oracle Privileged Account Manager Console, command line tool, or REST API.

Note:

- For command line instructions, refer to Section A.5, "Working with Targets."
- For REST API instructions, refer to Section B.6, "Target Resource."
- The procedures for showing and resetting a privileged account password are different from the procedures described in this section. Refer to Section 9.8, "Managing Privileged Account Passwords" for information.

Oracle Privileged Account Manager audits password management actions to keep track of password access.

7.3.1 Showing Service Account Passwords

If necessary, you can review the stored password for a target's service account by using the **Show Password** option, located above the Search Results table on the Targets page.

Note:

- This command is not applicable for the lockbox target type and it will return an "`Operation not supported`" error message.
- If someone changes a target's service account password from a location other than the current Oracle Privileged Account Manager instance, such as from another Oracle Privileged Account Manager instance in a different domain, the **Show Password** feature cannot display the new password and connections to the target will fail.

  To resolve this situation, you must update the password in Oracle Privileged Account Manager by editing the target from the Console or from the command line.

Use the following steps:

1. Select **Targets** in the Administration accordion.
2. When the Targets tab is displayed, use the Search portlet to locate the target.
3. Select the target row number and then click **Show Password**.

   The Show Current Password dialog box is displayed and it provides the following information about the target's service account password:

   - Target Name
   - Service Account Name
   - Current Password
   - Password Change Time
4. When you are finished, click **Close**.

7.3.2 Viewing the Password History

Use the **Password History** option to view the password history for a target's service account.

Note: Password History is not available for lockbox targets.

To view a target's password history,

1. Select **Targets** in the Administration accordion to open the Search Targets page, and then click **Search**.
2. Select the row number of the target.
3. When the **Password History** icon becomes active, click **Password History.**

   The Show Password History dialog box is displayed with the Target Name, and the Password in clear text, and the Modification Time (date and time of the password reset).
4. When you are finished click **Close**.

7.3.3 Resetting Service Account Passwords

If necessary, you can manually reset the stored password for a target's service account by using the **Reset Password** option, located above the Search Results table.

> **Note:** The **Reset Password** option is not applicable for the lockbox target type or the ldap target type and, if selected, it will return an "`Operation not supported`" error message.

Use the following steps:

1. Select **Targets** in the Administration accordion.
2. When the Targets tab is displayed, use the Search portlet to locate the target.
3. Select the target row number and then click **Reset Password**.

   The Reset Password dialog box is displayed and provides the following information about the target's service account password:

   - Target Name
   - Service Account Name

   This dialog box also contains two options for resetting the password:

   - **New Password**: Type a new password into the space provided.
   - **Generate password automatically**: Enable the checkbox to automatically generate a password, according to the account's Password Policy.
4. Type a new password or enable the checkbox, and then click **Reset**.

7.3.4 Understanding Service Account Password Rollover

In Oracle Privileged Account Manager, the service account for a target is governed by the password policy assigned to the target.

Password rollover for a target's service account is similar to password expiration for privileged accounts. If you enable password rollover for the service account, and the password has not been changed by the expiration date configured in the associated Password Policy, then Oracle Privileged Account Manager will automatically change the password to a randomized value.

> **Note:** Refer to Section 6.2, "Adding and Configuring Targets in Oracle Privileged Account Manager" for information about enabling password rollover for the different target types.

8

Configuring and Managing Agents

This chapter describes how you can configure and manage agents to work with Oracle Privileged Account Manager. The procedure to do so is described in the following sections:

- Section 8.1, "What is an Agent?"
- Section 8.2, "Deploying the OPAM Agent on a Windows Target"

8.1 What is an Agent?

An agent is a specifically designed tool that is deployed on a target, which is configured to perform a particular set of actions such as recording user actions. This section discusses the following topics:

- Section 8.1.1, "What is an Oracle Privileged Account Manager Agent for Windows?"
- Section 8.1.2, "Architecture and Functionality of the OPAM Agent"

8.1.1 What is an Oracle Privileged Account Manager Agent for Windows?

In Oracle Privileged Account Manager, all actions that are performed during a session checkout can be monitored using the Oracle Privileged Account Manager session monitoring feature. This feature records all the activities that a user performs during the privileged session checkout.

In this context, the Oracle Privileged Account Manager Agent for Windows targets (OPAM Agent) works specifically with Windows targets to enable Oracle Privileged Account Manager to monitor the actions performed by a user on a Windows target, during session checkout.

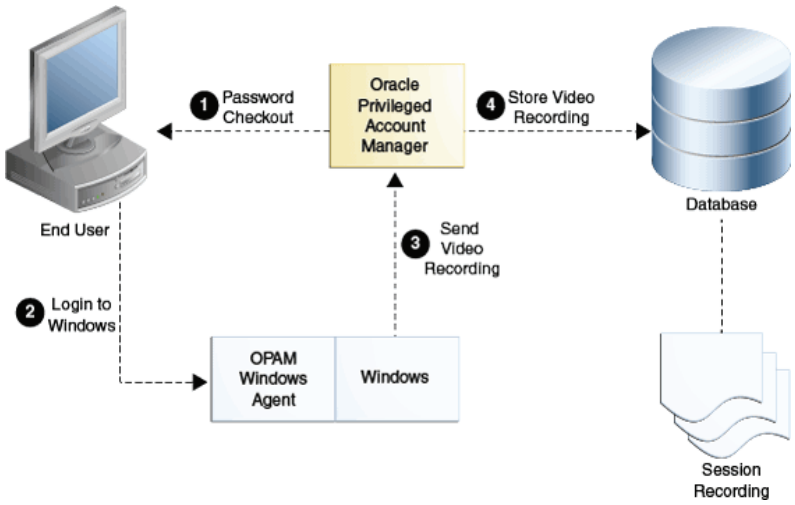
8.1.2 Architecture and Functionality of the OPAM Agent

The OPAM Agent is deployed directly on the Windows target. This agent runs the "OPAMAgentService" windows service on the target. This service then uses the "OpamAgentCapturer" child process to record user actions on the target. The service then converts the user actions into a video format and sends it securely to the Oracle Privileged Account Manager server periodically.

The OPAM Agent also sends metadata corresponding to the user's activity to the Oracle Privileged Account Manager server. The video data is saved into a database on the server. The metadata enables quick retrieval of relevant session recording videos. The playback for recorded videos is supported on HTML-5 compliant versions of the Chrome, Mozilla Firefox, Internet Explorer, and Safari browsers.

Figure 8–1 shows the end-to-end flow of session recording using the OPAM Agent.

Figure 8–1 End-to-End Flow of Session Recording

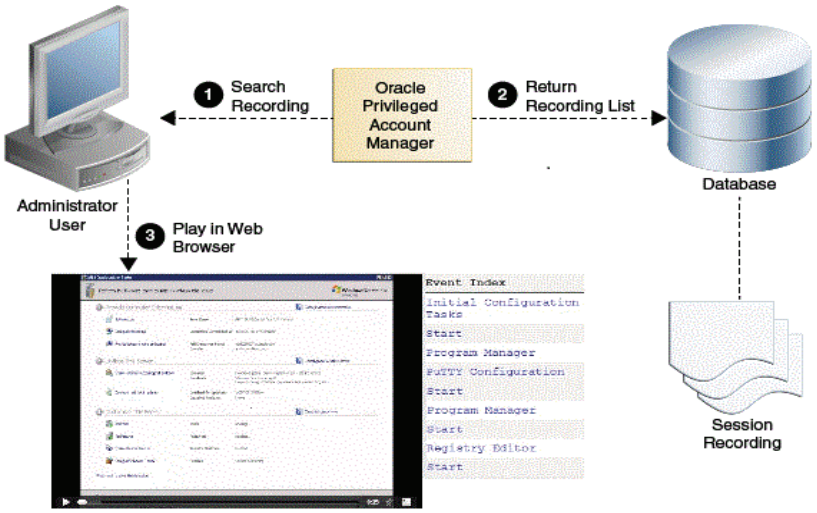


This figure shows the end-to-end flow of the session recording process enabled by the OPAM Agent.

****

Figure 8–2 shows the session recording replay flow for videos recorded using the OPAM Agent.

Figure 8–2 Session Recording Replay Flow



This figure shows the replay flow of session recordings that were recorded using the OPAM Agent.

****

8.2 Deploying the OPAM Agent on a Windows Target

The following sections describes the procedure to deploy the OPAM Agent on a Windows target:

Note: This procedure to deploy an OPAM Agent on a Windows target that is described in the following sections, assumes that the you have the following account-accesses and information:

- Administrator privileges on the Windows machine.
- Security Administrator privileges on the Oracle Privileged Account Manager Server where the Windows target needs to be registered.
- The OPAM Server URL.
- The Connector Server name as configured in the Oracle Privileged Account Manager Server.

8.2.1 Reviewing the Supported Components and Important Notes for Installation

The OPAM Agent is supported on the following Operating Systems:

Table 8–1 Supported Components

Component	Requirement
Microsoft Windows Client Operating System	You can use one of the following versions of the Microsoft Windows Client Operating System: ■ Microsoft Windows 2008 R2 ■ Microsoft Windows 7 ■ Microsoft Windows 8 ■ Microsoft Windows 8.1
Microsoft Windows Server Operating System	You can use one of the following releases of the Microsoft Windows Server Operating System: ■ Microsoft Windows 2012 Server ■ Microsoft Windows 2012 Server R2
.NET Version	4.5 or above

8.2.1.1 Important Notes for Installation on Microsoft Windows Server

This section is applicable while installing the OPAM Agent on the Microsoft Windows 2008 R2, Microsoft Windows 2012 Server, and Microsoft Windows 2012 Server R2 targets. The procedure outlined below has to be performed prior to installing the OPAM Agent and might require a restart of the system.

Note: No special preinstallation steps are needed for Microsoft Windows 7, Microsoft Windows 8, and Microsoft Windows 8.1

8.2.2 Setting Up the Windows Server

This section describes the actions you must perform on the Windows Server, before you begin to install the OPAM Agent. It describes the following topics:

- Section 8.2.2.1, "Enabling Desktop Experience for Microsoft Windows Server 2008 R2"
- Section 8.2.2.2, "Enabling Media Foundation Components For Microsoft Windows 2012 Server and Microsoft Windows 2012 Server R2"

8.2.2.1 Enabling Desktop Experience for Microsoft Windows Server 2008 R2

Depending on how you choose to enable the Desktop Experience, perform one of the following procedures to enable the Desktop Experience on the Microsoft Windows 2008 R2 operating system.

Using the Initial Configuration Tasks Wizard

1. In the Customize This Server section, click **Add features.**
2. Select the **Desktop Experience** check box and click **Next.**
3. Complete the wizard by clicking **Install.**

Using the Server Manager

1. Open the Server Manager and click **Start.** Navigate to Administrative Tools, and click **Server Manager.**

   Note: You can also open Server Manager by typing the following at a command prompt:

   ```
   servermanager.msc
   ```

2. In the Features Summary section, click **Add features.**
3. Select the **Desktop Experience** check box and click **Next.**
4. Complete the wizard by clicking **Install.**

8.2.2.2 Enabling Media Foundation Components For Microsoft Windows 2012 Server and Microsoft Windows 2012 Server R2

You must install Media Foundation components on Microsoft Windows Server 2012 and Microsoft Windows Server 2012 R2 as described in the following procedure:

1. Open the Server Manager and click **Start.** Navigate to Administrative Tools, and click **Server Manager.**

   Note: You can also open Server Manager by typing the following at a command prompt:

   ```
   servermanager.msc
   ```

2. In Server Manager, navigate to the "Add Roles and Features" wizard. Continue to click the **Next** button in the wizard, till you reach "Select installation type."
3. In the "Select installation type" step, select **Role-based or feature-based installation,** and click **Next.**
4. In the "Select destinations server" step, select **Select a server from the server pool.** Choose the desired machine and click **Next.**
5. In the "Select features" page, select **Media Foundation** and click **Install.**
6. Restart the server.

> **Note:** This feature requires to be configured initially. Once the configuration is complete, restart the server for the changes to take effect.

8.2.3 Installing the OPAM Agent

The OPAM Agent for Microsoft Windows is packaged as a binary installer named "OPAMAgentInstaller.msi." This is a standard Microsoft Windows installer.

Perform the following procedure to install the OPAM Agent:

1. Copy the "OPAMAgentInstaller.msi" installer to the Windows host from the following location:

   ```
   $ORACLE_HOME/opam/tools
   ```

2. Double-click **OPAMAgentInstaller.msi** (the installer) to run it.
3. In the installation wizard, read the License Agreement and click **Next.** This will install the OPAM Agent binaries into the following location:

   ```
   C:\Program Files\OPAMAgent
   ```

> **Note:** You can also install OPAMAgentInstaller.msi by typing the following at a command prompt:
>
> ```
> msiexec /i OpamAgentInstaller.msi
> ```

8.2.4 Setting up the OPAM Agent

This section discusses the following topics:

- Section 8.2.4.1, "Registering the OPAM Agent with the Oracle Privileged Account Manager Server"
- Section 8.2.4.2, "Updating the Target Key in Oracle Privileged Account Manager"

After installation, you must set up the OPAM agent. The "OpamAgentUtility.exe" file is used to setup the OPAM agent.

You must have "Administrator" privileges on the system within which you want to deploy the agent. Navigate to the following location from the command prompt:

```
C:\Program Files\OPAMAgent\
```

This location contains the "OpamAgentUtility.exe" file. This executable program can perform the following actions:

Note: Depending on the action you want to perform, run one or more of the commands described in this section.

- To register the OPAM Agent, you must run the `OpamAgentUtility.exe -r` command.
- To update the client key into the Oracle Privileged Account Manager Server, you must run the `OpamAgentUtility.exe -u` command.
- Run the `OpamAgentUtility.exe -d` command only when you want to deregister the OPAM Agent.

- **Registering the OPAM Agent:** When you run the "OpamAgentUtility" with the `-r` option as described in the following sample code, the OPAM Agent is registered:

  Sample command: `OpamAgentUtility.exe -r`

  For more information about registering the OPAM Agent, refer to Section 8.2.4.1, "Registering the OPAM Agent with the Oracle Privileged Account Manager Server."

- **Updating the key of the Target in the Oracle Privileged Account Manager server:** When you run the "OpamAgentUtility" with the `-u` option as described in the following sample code, the auto-generated key of the Windows target is updated in the Oracle Privileged Account Manager server.

  Sample command: `OpamAgentUtility.exe -u`

  For more information about updating the key of the target, refer to Section 8.2.4.2, "Updating the Target Key in Oracle Privileged Account Manager."

- **Deregistering the OPAM Agent:** When you run the "OpamAgentUtility" with the `-d` option as described in the following sample code, the OPAM Agent is deregistered.

  Sample command: `OpamAgentUtility.exe -d`

  For more information about uninstalling and deregistering the OPAM Agent, refer to Section 8.2.7, "Un-installing and Deregistering the OPAM Agent."

Running the `OpamAgentUtility.exe` command without any options will list the usage information for this executable.

The logging information from this executable file is available in the following location:

`C:\ProgramData\Opam\OpamAgentUtility_`*`Year_Month_Day_Hour_Minute_Second`*`.log`

In this location, "Year_Month_Day_Hour_Minute_Second" is a placeholder text in the name of the log file. It represents the format of the timestamp at which the log file was created.

8.2.4.1 Registering the OPAM Agent with the Oracle Privileged Account Manager Server

Before using the OPAM Agent on the Target, you must register the Agent with the Oracle Privileged Account Manager server.

To register the Agent:

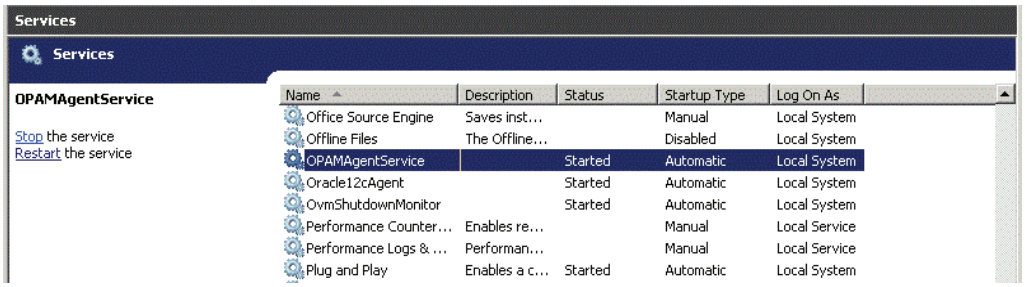
1. Run the `OpamAgentUtility.exe -r` command in the command prompt. The executable program will prompt for credentials to proceed with the registration.

2. To check for credentials, run the `OpamAgentUtility.exe` command to refer the usage information.

   The credentials can be provided using the interactive query or as command line arguments, as described in the usage information.

The executable program will start the OPAM Agent on the Windows target after it has successfully registered with the Oracle Privileged Account Manager server. If the registration is unsuccessful, check the log files as described in Section 8.2.4, "Setting up the OPAM Agent."

If the OPAM Agent was installed successfully, the service manager window will show the status of the "OPAMAgentService" service as "started." This is illustrated in the following screenshot:



If the Windows target, on which the OPAM Agent was deployed, is configured in Oracle Privileged Account Manager, then the registration process will automatically associate the agent with the specified target.

8.2.4.2 Updating the Target Key in Oracle Privileged Account Manager

The OPAM Agent uses an auto generated key to secure communication with the Oracle Privileged Account Manager server. You can update the key of the OPAM Agent to recreate a new auto-generated key.

> **Note:** Before you update the key, you must check if the Windows target, on which the OPAM Agent was configured, has been added to Oracle Privileged Account Manager.
>
> If this target has not been added to Oracle Privileged Account Manager, you must manually add the target. To do so, refer to Section 6.2, "Adding and Configuring Targets in Oracle Privileged Account Manager" for more information.

Perform the following procedure to update the Windows target key in Oracle Privileged Account Manager:

1. Open the "Command Prompt" as an "Administrator" on the system and navigate to the following location:

   ```
   C:\Program Files\OPAMAgent\
   ```

2. Run the following command and provide the necessary credentials to update the key of the target into the Oracle Privileged Account Manager server:

   ```
   OpamAgentUtility.exe -u
   ```

8.2.5 Logging Information for OPAM Agent

This section discusses logging information for the OPAM Agent. For information about Runtime Logs and Register-Time Logs, refer the following sections:

- Section 8.2.5.1, "Runtime Logs"
- Section 8.2.5.2, "Register-Time Logs"

The following is the primary log location:

```
C:\ProgramData\Opam
```

> **Note:** The preceding location is referred to as the "OPAM log folder" in this section.

The OPAMAgentService writes into the Windows Event History and this log is called "MyNewLog". It can be viewed using the Windows Event Viewer.

8.2.5.1 Runtime Logs

A directory is created in the OPAM log folder, for each checked-out session. The directory is named after the "username" of the user who checks-out the session. The runtime logs for these actions are stored in the following location:

C:\ProgramData\Opam\USERNAME\logs

Runtime logs are maintained for the following executables:

- OPAMAgentService.exe

  These logs are named in the `OpamAgentService_Year_Month_Day_Hour_Minute_Second.log` format.

  In this format, "Year_Month_Day_Hour_Minute_Second" represents the format of the timestamp at which the log file was created.

- OpamAgentCapturer.exe

  These logs are named in the `OpamAgentCapturer_Year_Month_Day_Hour_Minute_Second.log` format.

  In this format, "Year_Month_Day_Hour_Minute_Second" represents the format of the timestamp at which the log file was created.

8.2.5.2 Register-Time Logs

Register-time logs are logs for the actions associated with the "OpamAgentUtility.exe" program. These logs are also stored under the OPAM log folder. Register-time logs are named in the following format:

```
OpamAgentUtility_Year_Month_Day_Hour_Minute_Second.log
```

In this format, "Year_Month_Day_Hour_Minute_Second" represents the format of the timestamp at which the log file was created.

Logs for uninstallation or deregistration, and the OPAM Agent key update are also stored in register-time logs.

> **Note:** Because of format of the log file, the logs from registration and de-registration of the OPAM Agent could be in different log files.

8.2.6 Monitoring the End-to-End Flow of the Session Recording Process

You can monitor the end-to-end flow of the session recording process in the following situation, if the following actions are performed in sequence:

1. In Oracle Privileged Account Manager, an end user who is granted access to the Windows account, checks-out the password for the Windows account.
2. The end-user then logs in to the Windows target using the checked-out password.
3. The end user then performs certain actions on the Windows targets and logs out.

In the described situation, all session activity is now recorded as a video and stored securely on the Oracle Privileged Account Manager server. You can monitor the actions performed during this session checkout using the Checkout History Reports page from the console. Refer to Section 15.5, "Working with Checkout History Reports" for detailed information.

Note: In addition, any other sessions started directly on the Windows target without checking out the password from Oracle Privileged Account Manager will also be recorded by the OPAM Agent, and can be viewed in the Checkout History Reports page.

The value for the "username" column will show as `None` in the Checkout History Reports table for such sessions.

8.2.7 Un-installing and Deregistering the OPAM Agent

You can uninstall the OPAM Agent from the target. This will remove any run-time data (except logs) and remove the binaries stored in the following location:

```
C:\Program Files\OPAMAgent\
```

Uninstalling the OPAM Agent

Perform the following procedure to uninstall the OPAM Agent:

1. Login to the Windows target as an Administrator.
2. Navigate to the Control Panel and click **Add or Remove Programs.**
3. Select **OPAMAgent** from the list and click **Uninstall.** Follow through the steps in the wizard to complete the un-installation process.

Note: You can also uninstall the "OPAMAgentInstaller.msi" by typing the following in a command prompt:

```
msiexec /x OpamAgentInstaller.msi
```

Deregistering the OPAM Agent

You can also deregister the OPAM Agent without un-installing it from the target. Perform the following procedure to do so:

1. Login to the Windows target as an Administrator.
2. Open a command prompt and navigate to the following location:

   ```
   C:\Program Files\OPAMAgent\
   ```

3. To complete the deregistration process, run the `OpamAgentUtility.exe -d` command and provide values for the prompted parameters.

Note: The deregistration process will only remove the run-time data as described in this section.

9

Working with Privileged Accounts

This chapter provides some background information about privileged accounts and describes how to work with those accounts using the Oracle Privileged Account Manager Console.

This chapter includes the following sections:

Note: You can also manage Oracle Privileged Account Manager accounts from the command line or by using Oracle Privileged Account Manager's RESTful interface.

- For information about using the Oracle Privileged Account Manager Command Line Tool (CLI), refer to Section A.6, "Working with Accounts" in Appendix A, "Working with the Command Line Tool."
- For information about using the Oracle Privileged Account Manager RESTful interface, refer to Section B.7, "Account Resource" in Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface."

9.1 What is a Privileged Account?

An account on a target is considered *privileged* in a deployment when that account

- Is associated with elevated privileges
- Is used by multiple end-users on a task-by-task basis

- Requires its usage to be controlled and audited

You cannot create accounts in, or delete accounts from, your environment by using Oracle Privileged Account Manager. Oracle Privileged Account Manager only manages existing accounts that were provisioned using other mechanisms.

When you "add" an account in Oracle Privileged Account Manager, you are creating a reference to that account. In effect, you are registering the account and asking Oracle Privileged Account Manager to manage it. When you "remove" the account from Oracle Privileged Account Manager, you are only removing the reference to that account.

Note: ***Administrators determine which accounts are privileged within a particular deployment, and they must configure Oracle Privileged Account Manager to manage those accounts.***

You must be an Oracle Privileged Account Manager administrator with the *Security Administrator* Admin Role to add and manage accounts.

Oracle Privileged Account Manager enables you to manage both system and application accounts.

This section contains the following topics:

- Section 9.1.1, "Managing System Accounts"
- Section 9.1.2, "Managing Application Accounts"
- Section 9.1.3, "Understanding Sharing Accounts"

9.1.1 Managing System Accounts

Oracle Privileged Account Manager's primary purpose is to manage privileged system accounts on a supported target system. Oracle Privileged Account Manager does not mandate what constitutes a privileged system account — it can manage any account on a target system. Administrators are responsible for identifying which accounts are privileged. A privileged account is typically a system account that allows a user to perform administration tasks.

Privileged accounts are suitable for management through Oracle Privileged Account Manager if they are used and shared by multiple individuals in the organization and administrators are required to track the use of these accounts.

Administrators perform the following steps to register an account as a privileged account to be managed by Oracle Privileged Account Manager:

1. Add the target to Oracle Privileged Account Manager (if this has not already been done). Refer to Section 6.2, "Adding and Configuring Targets in Oracle Privileged Account Manager" for instructions.
2. Add the identified privileged account to the target and assign a Password Policy. Refer to Section 9.2, "Adding Privileged Accounts into Oracle Privileged Account Manager" and Section 10.2.5, "Assigning Password Policies" for instructions.
3. Grant access to end users directly or by using LDAP roles/groups and assign a Usage Policy. Refer to Section 11.2, "Granting Accounts to Users" and Section 10.2.5, "Assigning Password Policies" for instructions.

9.1.2 Managing Application Accounts

Applications use application accounts to connect to target systems at run time. Traditionally, administrators set up these accounts once during installation and then they are forgotten. Consequently, application accounts can potentially cause hidden vulnerabilities in your deployment. For example, passwords might become less secure over time because they were created using outdated policies or commonly used deployment passwords might be compromised.

Oracle Privileged Account Manager enables you to better manage application accounts. In particular, for applications that store their application accounts in the Credential Store. These applications consume the account credentials at run time from the Credential Store through the Credential Store Framework.

For example, because an application account is essentially a special version of a system account, you can register an application account in Oracle Privileged Account Manager as described in Section 9.1.1, "Managing System Accounts." You can then add the corresponding CSF mappings for every application that depends on that account, which is how CSF uniquely identifies a credential stored within CSF, and how an application finds its credential in CSF. For more information about CSF mapping, refer to "Guidelines for the Map Name" in the *Oracle Fusion Middleware Application Security Guide*.

If you register an account's CSF mappings with Oracle Privileged Account Manager, then every time the account's password changes, Oracle Privileged Account Manager can update the CSF entries that correspond to the registered mappings to reflect the new password and the applications continue to work without service interruption.

Note: Oracle Privileged Account Manager updates, or synchronizes, CSF *only* when a password change occurs. Refer to Section 19.3, "Integrating with the Credential Store Framework" for information about integrating Oracle Privileged Account Manager with CSF.

You can also use the plug-in framework to synchronize passwords to non-CSF application wallets. You can write a plug-in on the `passwordcycle` and `resetaccountpassword` operations for the `Server` resource to capture all password update operations, and then add custom logic to synchronize the resource to your application wallet. Refer to Section 13.2.7, "Supported Operations and Timings" for more information.

Additionally, you can apply a Password Policy to these applications that periodically cycles the account password. Cycling the password ensures that the application accounts are always compliant with the latest corporate policies and they remain secure. Oracle Privileged Account Manager performs this task with no service interruption.

Finally, its useful to note that Oracle Privileged Account Manager can support an account as both a system account (shared and used by multiple end-users) and as an application account (only used by an application at run time) at the same time. In this configuration, a human end-user who's been granted access can "check out" the application account to perform manual administrative operations as that application without disrupting application functionality.

9.1.3 Understanding Sharing Accounts

Oracle Privileged Account Manager enables you to specify whether an account is *shared* or *not shared*.

- **Shared accounts** enable multiple users to check out the account at the same time.
- **Unshared accounts** (Default) enable only one user to check out an account at a time.

Because unshared accounts are more secure, Oracle recommends that you designate an account as shared only if there are compelling business reasons to do so. If sharing is necessary, be sure to read Section 2.4.2, "Securing Shared Accounts."

Note: If you configure a shared account, be aware that a user can still use the password after checking in the account. Oracle Privileged Account Manager does not reset the account password until the last user checks in the account.

This is a security limitation for shared accounts.

9.2 Adding Privileged Accounts into Oracle Privileged Account Manager

---

Note: Accounts are always added to a target, so you must add a target object before you can add an account. Refer to Section 6.2, "Adding and Configuring Targets in Oracle Privileged Account Manager" for more information.

Never use the same account as the service account *and* as a privileged account to be managed by Oracle Privileged Account Manager. Refer to Chapter 7, "Working with Service Accounts" for information about service accounts.

You can add a new privileged account from either of the following pages:

- From the Accounts Page
- From the Targets Page

From the Accounts Page

To add an account by using the Accounts page,

1. Select **Accounts** from the Administration accordion.
2. Click the **Add** icon located above the Search results table.

From the Targets Page

To add an account by using the Targets page,

1. Select **Targets** in the Administration accordion.
2. Click **Search** in the Search Targets portlet to populate the Search Results table with a list of all available targets.
3. Locate the target where you want to add the account and click the name of the target to open it. The target opens in a new tab.
4. Select the **Privileged Accounts** tab.

5. Click **Add** in the table toolbar.

From both the "Accounts" and "Targets" page, when you click **Add**, the "Account: Untitled" page is displayed with the following subtabs:

> **Note:** Only the General tab is active at this point.

- **General:** Use to specify information needed to add the account.
- **Grants:** Use to associate users and groups (*grantees*) with the account.
- **Credential Store Framework:** Use to add or remove Credential Store Framework (CSF) mappings for the account.
- **Checkout History:** Use to search for, and view information about, any users who check out this account. Refer to "From the Checkout History Tab" on page 9-18 for more information.
- **Member-of:** This tab contains a table listing the different resource groups, which the privileged account is a member of.

Use these subtabs and the instructions provided in the following sections to finish adding the account:

- Section 9.2.1, "Adding the Account"
- Section 9.2.2, "Adding Grantees"
- Section 9.2.3, "Adding CSF Mappings"

9.2.1 Adding the Account

To add an account you must complete the Step 1: Set Target and Step 2: Add Account sections on the General tab as follows:

Set the Target

1. Provide a Target Name and Target Type.
   - If a Target Name and a **Target Type** are already displayed, proceed to Step 1 in the Set the Account section.
   - If the either parameter is <*undefined*>, click the search icon.
2. When the Set Target dialog box is displayed, enter a value in the **Target Name** field and click the **Search** button to locate the target where you want to add the account.

   For example, if you know the target name begins with "r," you can type an **r** into the **Target Name** field and click the **Search** button.
3. When the search results display in the Search Results table, select (check) the **Row** box next to a target name and then click **Set**.

   The selected Target Name and its Target Type are displayed on the General tab.

Set the Account

1. If the **Account Name** field is blank, click the search icon.
2. When the Set Account dialog box is displayed, enter one or more letters in the **Account Name** field and click the **Search** button to locate the account you want to add.

Note: Wildcard searches (for example, using percent (%) or underscore (_) symbols) are not supported in the Set Account dialog box because you perform search account operations against real targets.

For example, if you know the account name begins with "s," you can type an **s** into the **Account Name** field and click the **Search** button.

Note: When you add privileged accounts to a lockbox target, a **Password** field is also displayed in the Console.

Oracle Privileged Account Manager does not manage accounts on lockbox targets; therefore it cannot reset the passwords on those accounts. You must provide the password to be used when users check out those privileged accounts.

For more information about lockbox targets, refer to Section 6.1, "What Are Targets?"

---

3. When the search results display in the Search Results table, select (check) the **Row** box next to an account name and then click **Set**.

Note: You must not add the target's service account as a privileged account to be managed by Oracle Privileged Account Manager.

The selected account is displayed as the Account Name on the General tab.

4. Enable the **Shared Account** box to allow multiple users to check out this account at the same time.

Note: Refer to Section 9.1.3, "Understanding Sharing Accounts" and Section 9.5, "Checking Out Privileged Accounts" for more information.

---

5. Specify a **Password Policy**.

Note: Oracle Privileged Account Manager automatically assigns the Default Password Policy to new accounts. However, Oracle Privileged Account Manager administrators with the *Security Administrator* or the *User Manager* Admin Role can create new policies.

You can leave the default policy set or choose a different policy from the **Password Policy** drop-down menu.

For more information about policies, refer to Chapter 10, "Working with Policies."

---

6. Click **Test** to confirm that the account can be managed by Oracle Privileged Account Manager with these settings.

   If the account configuration settings are valid, a "`Test Succeeded`" message is displayed.

7. Click **Save**.

> **Note:** The Grants, Credential Store Framework, and Checkout History tabs do not become active until you save the new account information.

A new Current Checkouts section is displayed at the bottom of the General tab page. The table in this section enables you to view the following:

- Which users currently have the account checked out
- Type of checkout (password or session)
- Checkout expiration date
- Recordings (or transcripts) related to the account checkout

In addition, if you are an administrator with the *User Manager* Admin Role, you can use the **Force check-in** option to check in accounts. Refer to Forcing a Check-In.

You can now add grantees and CSF mappings to the account. Continue to Section 9.2.2, "Adding Grantees" and Section 9.2.3, "Adding CSF Mappings" for more information.

9.2.2 Adding Grantees

This section provides instructions for adding grantees to a privileged account.

> **Note:**
>
> - You must be an Oracle Privileged Account Manager administrator with the *User Manager* Admin Role to add, edit, or delete grantees.
> - Adding a new account does not automatically grant you access to that account. You must complete the process for adding yourself as a grantee.
> - Before adding grantees to an account, be sure to read Section 2.4.4, "Avoiding Assignments through Multiple Paths."

To associate users and groups with a new account, select the Grants tab and then complete the following steps:

- To associate users, click **Add** from the Users table toolbar.
  1. In the Add Users dialog, enter one or more letters of a name into the **User Name** field and click the arrow icon to search for that user.
  2. When the search results display, select (check) each user you want to associate with this account.
  3. When you are finished adding users, click **Add** and then click **Close**.

     Oracle Privileged Account Manager adds those user names to the Users table on the Grants tab and automatically assigns the Default Usage Policy.

     To assign a different policy, select it from the **Usage Policy** menu.
- To associate groups, click **Add** from the Groups table toolbar.

1. In the Add Group dialog, enter a name into the **Group Name** field and click the arrow icon to search for that group.
2. When the search results display, select (check) each group you want to associate with this account.
3. When you are finished adding groups, click **Add** and then click **Close**.

   Oracle Privileged Account Manager adds those group names to the Groups table on the Grants tab and automatically assigns the Default Usage Policy.

   To assign a different policy, select it from the **Usage Policy** menu.

Note: Removing Grants

Removing a user or group grant from an account *does not* automatically cancel all existing checkouts.

When grantees check out an account, they are guaranteed access to that account until one of the following events occur:

- The grantee checks in the account
- Oracle Privileged Account Manager automatically checks in the account because the checkout duration has exceeded the expiration period specified by the account's Usage Policy
- An administrator forces an account check-in

However, after the account is checked in, the grantee cannot check out that account again unless an administrator re-adds them as a grantee.

Note: After adding grantees to a privileged account, refer to Chapter 11, "Working with Grantees" for information about how to grant accounts, how to search for and open grantees, and how to remove grantees from accounts.

9.2.3 Adding CSF Mappings

Oracle Privileged Account Manager enables you to securely store and synchronize account credentials with the Oracle Credential Store Framework (CSF). This capability is useful for managing the lifecycle of application passwords stored in CSF.

When you configure CSF synchronization for an account, Oracle Privileged Account Manager changes the account password based on the assigned Usage Policy.

Note: Oracle Privileged Account Manager updates, or synchronizes, CSF *only* when a password change occurs.

For more information about CSF and how Oracle Privileged Account Manager manages CSF credentials, refer to Section 19.3, "Integrating with the Credential Store Framework."

To add CSF mappings to an account, complete the following steps:

1. Select the Account Name link in the Search Results table.
2. When the "Account: *AccountName*" page is displayed, select the Credential Store Framework tab.

3. Click **Add**.

   A new row is displayed in the table with empty fields in each column.

4. Enter the following information into the empty fields:

   - **Administration Server URL**. Enter the server URL in this format, *protocol*://*listen-address*:*listen-port*

     For example, if you are using the https protocol and the SSL port is 7002, you would enter

     `https://`*localhost*`:7002`

   - **Username** and **Password**. Enter the login credentials of the Oracle WebLogic Server administrator.
   - **Mapping**. Enter the Map name you created in CSF.
   - **Key**. Enter the unique Key you created in CSF.

5. Click **Add** again to create another mapping. You can create as many CSF mappings as needed.

6. When you are finished adding information, click **Test** to validate the mapping.

   A dialog box is displayed with either a success message or an error message.

9.3 Searching for Privileged Accounts

You can search for accounts by using one or more of the following parameters:

- Account Name
- Target Type (All, ldap, lockbox, database, sapum, sapume, ssh, unix, or windows)
- Target Name
- Privilege (All, security_admin, user_manager, or so on)

  Users can search accounts based on what privilege they have associated with those accounts.

- Domain
- Description
- Password Age

  Password age is defined as the number of days since the password was last changed. This is measured in days.

To search for an account,

1. Select **Accounts** in the Administration accordion.

2. When the Accounts tab is displayed, use the Search portlet parameters to configure your search.

   - For example, to search for a list of all accounts on a particular target, enter one or more letters of the target name into the **Target Name** field.
   - To search for all available accounts, do not specify any search parameters.

3. Click **Search**.

   Review your search results in the Search Results table.

> **Note:** You can use the **View** menu, located above the Search Results table, to manage how the search results are displayed in the table. Refer to the table in Section 4.3.6, "Working with a Search Results Table" for more information.

4. To perform another search, click **Reset**.

9.4 Opening Privileged Accounts

Opening an account enables you to view or edit the configuration parameters for that account.

You can open privileged accounts from any Search Results table containing an Account Name link. For example,

1. Select **Accounts** in the Administration accordion and click **Search**.
2. When the results display in the Search Results table, locate the account you want to open and perform one of the following actions:
   - Click the Account Name (an active link) in the Search Results table.
   - Select the account Row and then click **Open**.

The Account: *AccountName* page opens. From this page, depending on your Admin Role, you can view and configure account settings related to the associated target, grants, Credential Store Framework, and checkouts.

9.5 Checking Out Privileged Accounts

Oracle Privileged Account Manager enables grantees to check out an account in two ways:

- **Password Checkouts**: Enables grantees to access and check out granted account by using encrypted passwords.
- **Session Checkouts** (*on UNIX systems only*): Enables grantees to access and check out granted accounts without ever knowing the actual account credentials.

> **Note:** You can also use the Oracle Privileged Account Manager command line tool or the RESTful interface to check out accounts.
>
> - To use the command line tool, refer to Section A.6.4, "`checkout` Command."
> - To use the RESTful interface, refer to Section B.7.10, "Check Out an Account."

This section discusses the following topics:

- Section 9.5.1, "Checking Out Passwords"
- Section 9.5.2, "Clearing Copied Passwords From the Clipboard"
- Section 9.5.3, "Checking Out Privileged Account Sessions"

9.5.1 Checking Out Passwords

Any administrator or end user can check out a privileged account password if they have been granted access to that account. Refer to Chapter 11, "Working with Grantees" for more information.

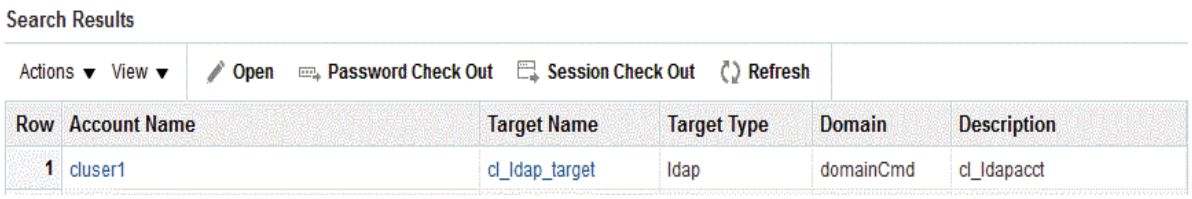
> **Note:** You must be an administrator with the *Security Administration* Admin Role to modify or remove an account.

Privileged accounts are *not shared* by default, which means when one user checks out the account, it becomes unavailable to other users and prevents conflicting actions. However, administrators can configure *shared* accounts, which enables multiple users to check out the account at the same time. Refer to Section 9.1.3, "Understanding Sharing Accounts" for more information.

Perform the following procedure to check out a password:

1. Select **My Accounts** in the Home accordion.
2. On the My Accounts page, locate the account you want to check out in the Search Results table and select that row as displayed below:

Figure 9–1 Account Available for Checkout



3. Click **Password Check Out**.
4. The Check-Out Account dialog box is displayed the Account Name, Target Name, and a field to enter justification. Enter a comment in this field if you choose to, and then click **Checkout**.

   If the checkout is successful, a Check-Out Account - Success dialog box is displayed.

   Depending on the user's password retrieval option that you have configured, perform one of the following procedures:

   > **Note:** Before you begin, refer to Step 3 of Section 5.2.3.1, "From the Console" for more information about configuring the Password Display option in the Server Configuration page.

   - To retrieve password for the "Enable copy password" option:

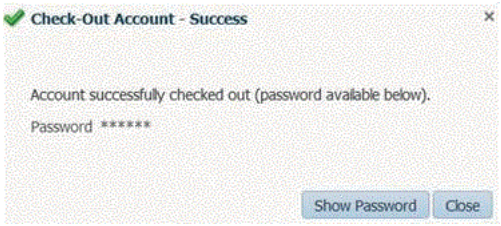
     In the Check-Out Account - Success dialog, click **Copy Password.**

     If the password is successfully copied, the "password copied" message is displayed. If the password copy is not successful, the "password not copied" message is displayed.

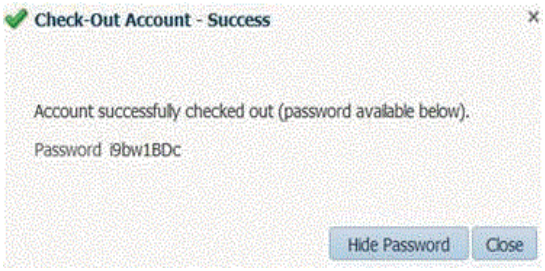
     You can hover over the info icon (i) beside the "password not copied" message for details about the workaround when the copy function fails.

- To retrieve password for the "Enable show password" option:

  In the Check-Out Account - Success dialog, click **Show Password,** as shown below:

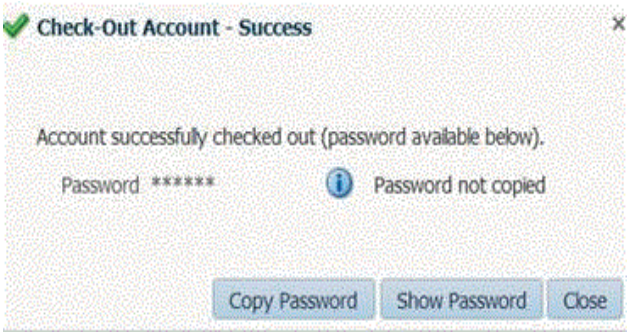


  The password is now displayed in plain text in the Check-Out Account - Success dialog box as shown below:



- To retrieve password for the "Enable show password and copy password" option:

  In the Check-Out Account - Success dialog, the **Copy Password** and the **Show Password** buttons are displayed as shown below:



  If you want to copy the password to a clipboard, click **Copy Password.** If you want to reveal the password in plain text, click **Show Password.**

If the checkout fails, the Check-Out Account dialog box is displayed with a message stating you cannot check out the account, which may indicate that someone else has already checked out that account.

Note: It is recommended that you clear the clipboard after using the password that you copied to the clipboard. For detailed information refer to Section 9.5.2, "Clearing Copied Passwords From the Clipboard."

Note: To see if an account is already checked out, click the account name. When the Account: *AccountName* page opens, you can review the Current Checkouts table to see who checked out the account, what type of checkout it was (password or session), when the account was checked out, the checkout expiration date, and view a recording (if available).

5. Click **Close** to close the dialog box and return to the Search Results table.
6. To verify that you checked out the account successfully:
   - Select **My Checkouts** from the Home accordion. When the My Checkouts page is displayed, locate the account name in the table.
   - If you have the *Security Administrator* Admin Role or the *User Manager* Admin Role, you can select **Accounts** from the Administration accordion and click **Search**. When the results are displayed, select the account name in the table to open the Account: *AccountName* page. The account should be listed in the Current Checkouts table.

9.5.2 Clearing Copied Passwords From the Clipboard

Oracle Privileged Account Manager allows you to clear any passwords you have copied to the clipboard.

Note: The app_config role administrator must set the "Password retrieval option" to either the "Enable copy password" or the "Enable show password and copy password" option to enable the "Clear Clipboard" functionality.

From the My Checkouts Page

You can access the **Clear Clipboard** option from the My Checkouts page as follows:

1. Select **My Checkouts** in the Home accordion to open the My Checkouts page.
2. In the My Checkouts page, click the **Clear Clipboard** option in the checkout results tool bar.
3. In the "Confirm Clear" message that is displayed on the screen, select **Clear.**

From the My Accounts Page

You can access the **Clear Clipboard** option from the My Accounts page as follows:

1. Select **My Accounts** in the Home accordion to open the My Accounts page.
2. In the My Accounts page, click the **Clear Clipboard** option in the "Search Results" tool bar.
3. In the "Confirm Clear" message that is displayed on the screen, select **Clear.**

9.5.3 Checking Out Privileged Account Sessions

Privileged sessions provide an extra level of security for privileged accounts on UNIX targets. Through privileged sessions, a grantee can access the granted account without ever knowing the actual account credentials.

Note: Session checkout is not available for other target types.

Any administrator or end user can check out a privileged account session if they have been granted access to that account and if the Usage Policy associated with the account allows session checkouts. Refer to Chapter 11, "Working with Grantees" and Section 10.3.4, "Modifying the Default Usage Policy" for more information.

To check out a session,

1. Select My Accounts in the Home accordion and then click **Search**.

   The My Accounts page is refreshed and all of your accounts are displayed in the Search Results table.

   Note: If you already know to how to establish the session using an SSH client, and you know the Oracle Privileged Account Manager server host, port, UNIX target, and UNIX account name, proceed to Step 3.

2. Select the account row, and then click **Session Check Out** to view the connection information you need to establish the session using an SSH client or securely transfer files using SCP.

   For example:

   ```
   Account Name:  opamuser1
   Target Name:   sample-unix
   SSH Port:      1222
   Instruction:   ssh -p <port> <opamuser>:<targetname>:
                  <accountname>@<sessionmgrhost>
                  Use opam password on password prompt.
   ```

   Where:

   - **port** is the port where Oracle Privileged Session Manager is running.
   - **opamuser** is the Oracle Privileged Account Manager end user.
   - **targetname** is the name of the target to which you are connecting.
   - **accountname** is the account you will be using on that target.
   - **sessionmgrhost** is the host on which you are running Session Manager.

   Note: The preceding example uses default Oracle Privileged Account Manager connection settings and instructions. Oracle Privileged Account Manager administrators can configure this information to whatever is appropriate for their own environments.

   Refer to Section 5.3, "Managing the Oracle Privileged Session Manager Server" for information about configuring these settings.

3. You can establish an SSH session or transfer files using SCP. Depending on your requirement, perform one of the following procedures:

   - To establish an SSH session:

     Use any favorite SSH client to connect to a target or an account through the Oracle Privileged Session Manager server.

For example, using the SSH client on a standard Linux machine, you would perform the following steps:

a. Open a command window.

b. At the prompt, enter the connection information as noted in the Session Checkout dialog.

For example:

```
scp -P 1222
"opamuser1;DemoMachineTarget;root"@sessionmgr.example.com:foobar.txt
/some/local/directory
```

A message is displayed stating that you are authenticated with partial success.

- To transfer files using SCP:

  Use your favorite SCP client to connect to a target or an account through the Oracle Privileged Session Manager server to securely transfer files.

  For example, using the SCP client on a standard Linux machine, you would perform the following steps:

  a. Open a command window.

  b. At the prompt, enter the following connection information:

  ```
  scp -P <port>
  "<opamuser>;<targetname>;<accountname>"@<sessionmgrhost>:src_path
  des_path
  ```

  For example:

  ```
  prompt> scp -p 1222
  opamuser1;target_system;user1@sessionmgrhost:examplefile
  /example/directory
  ```

  A message is displayed stating that you are authenticated with partial success.

4. Enter the appropriate Oracle Privileged Account Manager password when you see the prompt to complete the connection to the Oracle Privileged Session Manager server.
5. To confirm the connection, type `id` at the prompt, and the account's `uid`, `gid`, and `group` information will be returned.
6. Return to the My Accounts page in the Console
7. To verify that you checked out the session successfully:
   - Select **My Checkouts** from the Home accordion. When the My Checkouts page is displayed, locate the account name in the table and review the Checkout Type column.
   - If you have the *Security Administrator* Admin Role or the *User Manager* Admin Role, you can select **Accounts** from the Administration accordion and click **Search**. When the Search Results display, select the account name in the table to open the Account: *AccountName* page. The session should be listed in the Current Checkouts table.

Note: You do not have to perform any special steps to check in a checked out session. If you use the procedure described in Section 9.6, "Checking In Privileged Accounts," then the account is checked back in regardless of the checkout type (password or session).

9.6 Checking In Privileged Accounts

Any administrator or end user can check in their checked-out accounts by using the steps described in Regular Check-In.

Administrators with the *User Manager* Admin Role can *force* an account check-in (check in privileged accounts that have been checked out by other users) when necessary. Use the steps described in Forcing a Check-In.

Note: In either case, you use the same steps to check in an account password or an account session.

Regular Check-In

To check in a checked out privileged account:

1. Select **My Checkouts** on the Home accordion.

   The My Checkouts page is displayed with all of your checked-out accounts (passwords and sessions) listed in the Search Results table.

2. Select the account row or rows you want to check in.
3. When the **Check-in** icon located above the table becomes active, click the icon.
4. When the Check-in Accounts dialog box is displayed, click the **Check In** button.

   If the check-in is successful, Oracle Privileged Account Manager removes the account name(s) from the My Checkouts table and the account becomes available for check out again.

Forcing a Check-In

To force an account check in:

1. Select **Accounts** in the Administration accordion, and then search for the account as described in Section 9.3, "Searching for Privileged Accounts."
2. Select (check) the account you want to check in.
3. When the **Force Check In** option located above the table becomes active, click the icon.

   The Confirm Forced Check In dialog box is displayed, asking you to confirm that you want to check in the account. Be aware that forcing the check in will log out all users that currently have the account checked out.

4. To proceed, click the **Check In** button.

   If the check-in is successful, the account becomes Available for check out again.

Note: You can also use the Oracle Privileged Account Manager command line tool or the RESTful interface to check-in accounts.

- To use the command line tool, refer to Section A.6.3, "`checkin` Command."
- To use the RESTful interface, refer to Section B.7.15, "Check In an Account."

9.7 Viewing a Session Recording

If necessary, administrators can view a recording in plain text transcript format, interactive transcript format, and video format.

Note: The **Session Monitoring Update Interval in seconds** setting on the Session Manager Configuration Page controls how often on-going session transcripts are updated. Refer to step 2 of the procedure in Section 5.3.3, "Managing the Oracle Privileged Session Manager Properties" for more information.

The following table describes the different transcript types, where you can access these recordings, and also provides information about which Admin Roles are required to view the transcripts:

Recording Type	Viewing Location	Admin Role
On-going session transcripts	The Current Checkouts table of the account	Security Admin or User Manager **Note:** A delegated security admin or a delegated User Manager can also view this recording
Expired session transcripts	The Checkout History tab of the account	Security Admin or User Manager **Note:** A delegated Security Admin or a delegated User Manager can also view this recording
Expired session transcripts	Checkout History Report page	Security Auditor
Expired session video	Checkout History Report page	Security Auditor

The following sections provide instructions for accessing these recordings:

From a Current Checkouts Table

1. Open the account as described in Section 9.4, "Opening Privileged Accounts."
2. When the "Account: *AccountName*" page is displayed, locate the correct user in the Current Checkouts table, and click the **Recording** icon in that row.

   A new tab opens in your browser and the recording is displayed in one of the following formats: a transcript format. For example,

- For SSH recording: A transcript of the user's actions is displayed. Refer to Chapter 15.5, "Working with Checkout History Reports" for more information.
- For Windows Session recording: A video recording of the user's actions is displayed. Refer to Chapter 15.5, "Working with Checkout History Reports" for more information.

From the Checkout History Tab

1. Open the account as described in Section 9.4, "Opening Privileged Accounts."
2. When the "Account: *AccountName*" page is displayed, select the Checkout History tab.

   A new tab opens in your browser and the recording is displayed in a transcript format.
3. Specify a period during which the search can be performed. This can be done by setting values for the **Start Date** and **End Date** (*required*) fields. Type the date and time into the blank fields or use the **Calendar** icons.

   Include any other, optional search criteria in the Search Checkout History section, and then click **Search**.
4. When the search results display in the table, locate the user whose transcript or video that you want to review, and click the **Recording** icon in that row.

From the Checkout History Page

1. Select the **Checkout History** link from the Reports accordion to open the Checkout History page.
2. Use the Search Checkout History portlet to configure search parameters:
   - You must specify a **Start Date** and an **End Date** range in which to search for checkouts. Type the date and time into the blank fields or use the **Calendar** icons.
   - Enter information into one or more of the **Account Name**, **User Name**, **Target Name**, or **Pattern** fields.

     Note: Use the **Pattern** field to search for a string in the recording of a checkout event.

     If the recent session recordings are not listed in pattern search results, refer to Section 20.3.22, "Checkout History Search Results for Pattern Search Do Not Include Recent Session Recordings" to troubleshoot this issue.
   - Enter a value into the **Query** field to limit the number of returned results.
3. Click **Search** and the results will be displayed in the table.
4. Locate the correct account and user row in the table, and click the **Recording** icon in that row.
5. You are prompted to select a program, in which to open the recording. Select a program and click **Open**.

   The recording opens in the selected program, and is displayed in a transcript format or video format.

9.8 Managing Privileged Account Passwords

Oracle Privileged Account Manager provides the following options for managing privileged account passwords:

- Section 9.8.1, "Showing an Account Password"
- Section 9.8.2, "Viewing an Account's Password History"
- Section 9.8.3, "Resetting an Account Password"

Note: You can also perform these password management tasks by using the Oracle Privileged Account Manager command line tool or REST API.

- For command line instructions, refer to Section A.6, "Working with Accounts."
- For REST API instructions, refer to Section B.7, "Account Resource."

Oracle Privileged Account Manager audits password management actions to keep track of password access.

Note: The procedures for showing and resetting a target's *service account* password are different from the procedures described in this section. Refer to Section 7.3, "Managing Service Account Passwords" for information.

9.8.1 Showing an Account Password

If necessary, you can view a password in clear text for an account that you have checked out by using the **Show Password** option. For example, if you forget a password, you can use this feature to view the password again.

Any user can review passwords for accounts they have checked out. However, you cannot access passwords after the account is checked back in or view passwords for accounts that are checked out by other users. Attempts to do so will cause an error.

Note: Administrators with the *Security Administration* Admin Role, who can access all system and target service accounts, can use this feature to view current the password for both checked out and checked in privileged accounts.

From the My Checkouts Page

You can access the "Show Password" option from the "My Checkouts" page as follows:

1. Ensure that you have the privileged account checked out.

Note: For most users, if they try to view the password for an account that has already been checked back in, an error will result.

However, if you are an administrator with the *Security Administrator* or *User Manager* Admin Role, you can use this command to reset a password for both checked out and checked-in accounts.

---

2. Select **My Checkouts** in the Home accordion to open the My Checkouts page.
3. Select the row number of the account.
4. Click the **Show Password** icon when it becomes active.

   The current password dialog box will appear. Depending on the user's password retrieval option that you have configured, click the **Show Password** or **Copy Password** option. For detailed information about the Show Password and Copy Password actions, refer to Step 4 of the procedure in Section 9.5.1, "Checking Out Passwords."
5. Select the account row number and click the **Show Password** icon when it becomes active.

   A dialog box will then appear and display the account information and password.

From the Accounts Page

Administrators with the *Security Administration* or *User Manager* Admin Role can access the "Show Password" options as follows:

1. Select **Accounts** in the Administration accordion.
2. When the Accounts page is displayed, use the Search portlet to locate the account.
3. Select the account row number and click the **Show Password** icon when it becomes active. A dialog box will appear and display the account information and password.
4. When you have finished click **Close**.

9.8.2 Viewing an Account's Password History

Use the **Password History** option to view the password history for an account.

Note: You must be an administrator with the *Security Administration* Admin Role to view the password history for a privileged account.

To view a privileged account's password history,

1. Select **Accounts** in the Administration accordion to open the Search Accounts page, and then click **Search**.
2. Select the row number of the account.
3. When the **Password History** icon becomes active, click the icon.

   The Show Password History dialog box is displayed with the Account Name, and the Password in clear text, and the Modification Time (Date and time of the password reset).
4. When you are finished click **Close**.

9.8.3 Resetting an Account Password

If necessary, you can manually reset the existing password for an account that you have checked out by using the **Reset Password** option.

If Security Administrators do not want to use randomized password generation, they can manually set a password of their choosing. For example, administrators might

prefer to set a simple, easy-to-type password for one time use, such as during a system upgrade.

To reset an account password, use the following steps

1. Ensure that you have the privileged account checked out.

   Note: For most users, if they try resetting the password for an account that has already been checked back in, an error will result.

   However, if you are an administrator with the *Security Administrator* Admin Role, you can use this command to reset a password for both checked out and checked-in accounts.

2. Select **Accounts** in the Administration accordion.
3. When the Accounts tab is displayed, use the Search portlet to locate the account.
4. Select the account row number and then click **Reset Password**.

   The Reset Password dialog box is displayed and provides the following information about the account password:

   - Account Name
   - Target Name

   This dialog box also contains a **New Password** field.

5. Type a password into the space provided and click **Save**.

   You can use a password string of your choosing. The string does not have comply with the Oracle Privileged Account Manager Password Policy because the Password Policy is used for randomized password generation.

   A message is displayed with the name of the selected account and the new password.

9.9 Removing Privileged Accounts from Oracle Privileged Account Manager

You can remove a privileged account from Oracle Privileged Account Manager by using the Search Accounts page or the Targets page.

WARNING: When you remove a privileged account, you remove all information about the account that is stored in Oracle Privileged Account Manager.

Before removing a privileged account, it is critical that you first capture all relevant information from that account. For example, save the current password associated with that privileged account.

From the Search Accounts Page

To remove an account from the Search Accounts page,

1. Locate the account to remove.
   a. Select **Accounts** in the Administration accordion.

b. Click **Search** in the Search Accounts portlet to populate the Search Results table with a list of all available accounts.

   To narrow the results or to locate a particular account, enter search criteria in one or more the Search Accounts fields, and then click **Search**.

2. In the Search Results table, select the account to be removed, and then click **Remove**.
3. When you are finished, click the **Apply** button located at the top of the page.

From the Target Page

To remove an account from a target,

1. Locate the target from which you want to remove the account.
   a. Select **Targets** in the Administration accordion.
   b. Click **Search** in the Search Targets portlet to populate the Search Results table with a list of all available targets.

      To narrow the results or to locate a particular target, enter search criteria in one or more the Search Targets fields, and then click **Search**.
2. Click the target name in the Search Results table to open the target.
3. Select the Privileged Accounts tab.
4. In the Search Results table, select the account to be removed and then click **Remove**.
5. When you are finished, click the **Apply** button located at the top of the page.

10

Working with Policies

This chapter introduces Oracle Privileged Account Manager policies and describes how administrators can configure and manage policies from the Console.

This chapter includes the following sections:

- Section 10.1, "What Are Oracle Privileged Account Manager Policies?"
- Section 10.2, "Working with Password Policies"
- Section 10.3, "Working with Usage Policies"

> **Note:** Administrators can also manage Oracle Privileged Account Manager policies from the command line or by using Oracle Privileged Account Manager's RESTful interface. For information, refer to
>
> - Section A.4, "Working with Policies" in Appendix A, "Working with the Command Line Tool."
> - Section B.5, "Policy Resource" in Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface."

10.1 What Are Oracle Privileged Account Manager Policies?

In Oracle Privileged Account Manager, there are two types of policies:

- **Password Policies** define the password construction rules to be enforced by a specific target on an associated privileged account. This policy type also governs the password's lifecycle, or how often the password must be changed.

  For example, a Password Policy might require a minimum and maximum number of numeric characters in a password and require that a password must be changed every five days.

  You can also use a Password Policy to create passwords that enable Oracle Privileged Account Manager to reset the password for a privileged account.

- **Usage Policies** define when and how grantees can use a privileged account.

  You can also configure Usage Policies to constrain and enforce which tasks users are allowed to perform when they have session access.

Every privileged account that is managed by Oracle Privileged Account Manager must have an associated Password Policy. A Usage Policy only applies at the level of a grant. You can associate a single Password Policy with multiple privileged accounts and a single Usage Policy with multiple grants.

Note: For information about how grants are applied for Usage Policies, refer to Section 10.3.1.1, "Understanding How Grants are Applied."

Oracle Privileged Account Manager provides both a Default Password Policy and a Default Usage Policy. You can use these default policies, modify them, or create your own, specialized policies. Refer to Section 10.2, "Working with Password Policies" and Section 10.3, "Working with Usage Policies," respectively, for more information.

Note: You cannot delete the Default Password Policy or the Default Usage Policy.

Table 10–1 describes which Admin Roles can work with Oracle Privileged Account Manager policies and it describes which tasks each Admin Role can and *cannot* perform.

Table 10–1 Which Admin Roles Can Work with Policies

Administrators with this Admin Role	Can Perform this Task	Cannot Perform this Task
Security Administrator	■ Modify the Default Password Policy and Default Usage Policy ■ Create new Password Policies and Usage Policies ■ Delete Password Policies and Usage Policies ■ Assign Password Policies	Assign Usage Policies
User Manager	■ Assign a Usage Policy to accounts at the *grantee-account* pair level. In other words, the *User Manager* can assign different Usage Policies to different grantees of the same account.	Assign Password Policies

10.2 Working with Password Policies

This section describes the different tasks an administrator performs when working with Password Policies.

Note: You must be an Oracle Privileged Account Manager administrator with the *Security Administrator* Admin Role to work with and assign Password Policies.

The topics include:

- Section 10.2.1, "Searching for Password Policies"
- Section 10.2.2, "Viewing Password Policies"
- Section 10.2.3, "Modifying the Default Password Policy"
- Section 10.2.4, "Creating a Password Policy"
- Section 10.2.5, "Assigning Password Policies"
- Section 10.2.6, "Deleting Password Policies"

10.2.1 Searching for Password Policies

To search for a Password Policy,

1. Select **Password Policies** from the Administration accordion.
2. When the Search Policies portlet is displayed, enter your search criteria into one or more of the following fields.
   - **Name**: Enter all or any part of a policy name.
   - **Status**: Choose one of the following options from the menu.
     - Select **All** (*default*) to search for all policies (active and inactive).
     - Select **Active** or **Disabled** to limit the search to just active or just inactive policies.
3. Click **Search**.

The results are displayed in the Search Results table, which includes the Name and Status.

10.2.2 Viewing Password Policies

To review the parameter settings for a Password Policy,

1. Select **Password Policies** from the Administration accordion.
2. When the Policies page is displayed, click **Search**.

   The existing Password Policies are displayed in the Search Results table.
3. Use one of the following methods to open a policy:
   - Click the row number next to the name of the policy, and then click the **Open** icon located above the Search Results table.
   - Click the name (an active link) in the Search Results table.

     For example, clicking the **Default Password Policy** link opens the Password Policy: Default Password Policy page.

   A Password Policy page contains three tabs:

   - **General**. Select this tab to specify general information about the policy and to configure Password Lifecycle Rules for the policy.

     Password Lifecycle Rules govern when Oracle Privileged Account Manager must automatically reset an account password. Refer to Table 10–2, " Password Lifecycle Rules Parameters" for a description of these parameters.
   - **Password Complexity Rules**. Select this tab to set the rules that govern the complexity requirements for account passwords. Refer to Table 10–3, " Password Complexity Rules Parameters" for a description of these parameters.
   - **Privileged Accounts**. Select this tab to view information about the privileged accounts currently using the selected Password Policy.

10.2.3 Modifying the Default Password Policy

After evaluating the Default Password Policy, you may decide you want to modify the settings to better suit your environment.

Note: Oracle recommends making a back-up copy of the Default Password Policy before you make any changes. You can use the `export` command as described in Section A.10.1, "`export` Command."

To modify the Default Password Policy,

1. Select **Password Policies** from the Administration accordion.
2. When the Password Policies page is displayed, click **Search** to populate the Search Results table.
3. Click the **Default Password Policy** link in the Search Results table to open the Password Policy: Default Password Policy page.
4. Select the General tab to modify the **Description** in the General Fields area or to modify any of the following Password Lifecycle Rules:

Note: You cannot edit the **Name** or **Status** values for this policy.

Table 10–2 Password Lifecycle Rules Parameters

Parameter	Description
Save password history for	Use the counter and drop menus to specify how many days to save the password history for an account. The password history includes when accounts are checked out, checked in, and when their passwords were reset.
Expire password after	Use the counter and drop menus to specify a duration period (number of days, hours, or minutes) after which Oracle Privileged Account Manager must automatically reset the account password. For example, if your enterprise wants a security policy where account passwords must be changed every month, you would set this value to 30 days. Every time the account is checked out and its password gets changed (if the policy is configured so that passwords must be changed on checkout/check-in) Oracle Privileged Account Manager tracks the password change time. If Oracle Privileged Account Manager detects the account is idle and no password changes have occurred over the specified number of days, then Oracle Privileged Account Manager automatically resets the password to a new, randomized value, which helps the enterprise to automatically enforce the security policy without human intervention. To disable this automatic reset option, set the numeric value to 0. **Note**: The Oracle Privileged Account Manager scheduler periodically checks for accounts where the password maximum age has expired and resets them as described in this section. By default, the scheduler makes this check every 60 minutes (based on the `passwordcyclerinterval` property in the OPAM Global Config configuration entry, whose default setting is 60 minutes). You can view and modify the current interval by using Oracle Privileged Account Manager's `getconfig` and `modifyconfig` command line options. For more information, refer to Section A.2.1, "`getconfig` Command" and to Section A.2.3, "`modifyconfig` Command."
Reset password on check-in	Use this option to specify whether Oracle Privileged Account Manager must auto-generate and set a randomized password during a check-in operation. Uncheck this box if you do not want the password to be reset during the check-in operation.
Reset password on check-out	Use this option to specify whether Oracle Privileged Account Manager must auto-generate and set a randomized password during a checkout operation. Uncheck this box if you do not want the password to be reset during the checkout operation.

Note:

- An administrator with the *Security Administrator* Admin Role can also manually reset a password by using the **Reset Password** option (described in Section 9.8.3, "Resetting an Account Password") and Oracle Privileged Account Manager tracks this password change time as well.
- For higher security, the **Reset password on check-in** and **Reset password on check-out** options are both enabled by default, but they can be disabled if required. For example, some enterprises may only require that passwords be reset every 30 days.
- If your enterprise prefers that passwords not be automatically managed at all; that they are only changed through human intervention, disable all three Password Lifecycle Rules options.

  However, after disabling these three options, the only way to manually change passwords is by using the **Reset Password** option (described in Section 9.8.3, "Resetting an Account Password"). Oracle Privileged Account Manager is still useful in this case, as you can reset and centrally manage passwords for multiple systems from one place by using Oracle Privileged Account Manager.

5. Select the Password Complexity Rules tab to change one or more of the parameters that define the default password requirements.

Table 10–3 Password Complexity Rules Parameters

Parameter	Description
Characters for Password	Specify the minimum and maximum number of characters required.
Alphabetic Characters	Specify the minimum number of alphabetic characters required.
Numeric Characters	Specify the minimum number of numeric characters required.
Alphanumeric Characters	Specify the minimum number of alphanumeric characters required.
Special Characters	Specify the minimum and maximum number of special characters (such as `*` or `@`) required.
Repeated Characters	Specify the minimum and maximum number of repeated characters allowed.
Unique Characters	Specify the minimum number of unique characters required.
Uppercase Characters	Specify the minimum number of uppercase characters required.
Lowercase Characters	Specify the minimum number of lowercase characters required.
Start with Character (not digit)	Specify the first character required to start a password.
Required Characters	Specify which characters are required in a password.
Allowed Characters	Specify which characters are permitted in a password.
Disallowed Characters	Specify which characters are not permitted in a password.
Disallowed as Password	Enable (check) the **Account Name** box to prohibit the use of an account name in the password.

6. Select the Privileged Accounts tab to review which accounts are currently using the Default Password Policy.

Note: To specify a different Password Policy for any account listed in the table, click the **Account Name** link. When the Account page is displayed, select a different policy name from the **Password Policy** menu.

7. When you are finished editing the policy, click **Apply** to save your changes.

10.2.4 Creating a Password Policy

To create a Password Policy,

1. Select **Password Policies** from the Administration accordion.
2. When the Password Policies page is displayed, click **Create** at the top of the Search Results table.

   A new, "Password Policy: Untitled" page is displayed with three tabs.
3. Provide the following information on the General tab:
   a. Name: Enter a name for the new policy.
   b. Status: Click the **Active** or **Disabled** button to specify whether the policy is active or disabled.

      Making the policy Active puts that policy into effect for all of the associated accounts and grants.

      Disabling a policy applies the Default Password Policy to all accounts and grants associated with that disabled policy. If you simply assigned a different policy to those accounts and grants, you would lose all information about the old policy assignment.
   c. Description (*optional*): Enter a descriptive statement about the new policy.
   d. Password Lifecycle Rules: Configure these parameters to enable Oracle Privileged Account Manager to auto-generate and set a randomized account password under certain conditions. Refer to Table 10–2, " Password Lifecycle Rules Parameters" for more information.
4. Select the Password Complexity Rules tab to specify password complexity rules for this policy. Refer to Table 10–3, " Password Complexity Rules Parameters" for a description of these parameter settings.
5. Select the Privileged Accounts tab to assign the new policy to accounts or grantees. Refer to Section 10.2.5, "Assigning Password Policies" for detailed instructions.

   After assigning this Password Policy to privileged accounts, you can select the Privileged Accounts tab to review which accounts are currently using this policy.
6. Click **Save**.

10.2.5 Assigning Password Policies

When you add a new privileged account, Oracle Privileged Account Manager automatically assigns the Default Password Policy to that account. However, if you have created other Password Policies, as described in Section 10.2.4, "Creating a Password Policy," you can assign a different policy to the account.

Note: Only administrators with the *Security Administrator* Admin Role can assign Password Policies to accounts.

You can assign Password Policies to an account

- From the Accounts Page
- From the Targets Page
- From the Password Policies Page

From the Accounts Page

To assign a Password Policy from the Accounts page,

1. Locate the account where you want to assign the policy.
   a. Select **Accounts** in the Administration accordion.
   b. Click **Search** in the Search Accounts portlet to populate the Search Results table with a list of all available accounts.

      To narrow the results or to locate a particular account, enter search criteria in one or more the Search Accounts fields, and then click **Search**. For example, if you know the account is assigned to a UNIX target, select **unix** from the Target Type menu.
2. When the Search Results display, click the account's Account Name link in the table to open the Account: *AccountName* page.
3. On the General tab, select a different policy name from the **Password Policy** menu.
4. After selecting the new policy, click **Test** to verify that the account can be managed by Oracle Privileged Account Manager.

   If the test is successful, a "`Test Succeeded`" message is displayed.
5. Click **Apply** to finish assigning the policy to the selected account.

From the Targets Page

To assign a Password Policy from the Targets page,

1. Locate the target where the account is located.
   a. Select **Targets** in the Administration accordion.
   b. Click **Search** in the Search Targets portlet to populate the Search Results table with a list of all available targets.

      To narrow the results or to locate a particular target, enter search criteria in one or more the Search Targets fields, and then click **Search**.
2. Click the target name of the account (an active link) in the Search Results table to open the Target: *TargetName* page.
3. Click the **Privileged Accounts** tab to view a list of the accounts currently managed on the target.

   Notice that the table lists the Password Policy that is currently assigned to each account.
4. Locate the account in the Privileged Accounts table, and then click the name of the account, which is an active link.

5. When the General tab is displayed, select a different policy name from the **Password Policy** menu.
6. After selecting the new policy, click **Test** to verify that the account can be managed by Oracle Privileged Account Manager.

   If the test is successful, a "`Test Succeeded`" message is displayed.
7. Click **Apply** to finish assigning the policy to the selected account.

From the Password Policies Page

To assign a Password Policy from the Policies page,

1. Locate the Password Policy that you want to assign to the account.
   a. Select **Password Policies** in the Administration accordion.
   b. Click **Search** in the Search Policies portlet to populate the Search Results table with a list of all available Password Policies.

      To narrow the results or to locate a particular policy, enter search criteria in one or more the Search Policies fields, and then click **Search**.
2. Locate the policy in the Search Results table, and then click the name of the password policy (an active link) to open the "Password Policy: *PolicyName*" page.
3. Select the **Privileged Accounts** tab.
4. Locate the account and click the name of the account (an active link) to open the "Account: *AccountName*" page.
5. When the General tab is displayed, select a different policy name from the **Password Policy** menu.
6. After selecting the new policy, click **Test** to verify that the account can be managed by Oracle Privileged Account Manager.

   If the test is successful, a "`Test Succeeded`" message is displayed.
7. Click **Apply** to finish assigning the policy to the selected account.

10.2.6 Deleting Password Policies

> **Note:** You cannot delete the Default Password Policy.

To delete a Password Policy,

1. Locate and select the policy to be deleted.
2. Click **Delete.**
3. When the Confirm Remove dialog box is displayed, click **Remove.**

   The policy is immediately deleted. If you had any accounts assigned to that policy, they will all revert to using the Default Password Policy.

10.3 Working with Usage Policies

This section describes the different tasks an administrator performs when working with Usage Policies.

Note:

- You must be an Oracle Privileged Account Manager administrator with the *Security Administrator* Admin Role to work with (search, create, modify, or delete) Usage Policies.
- You must be an Oracle Privileged Account Manager administrator with the *User Manager* Admin Role to assign Usage Policies.

The topics include:

10.3.1 Before You Begin

Before you start working with Usage Policies, you should understand the concepts described in the following sections:

10.3.1.1 Understanding How Grants are Applied

A Usage Policy only applies at the level of a grant. For Usage Policies, Oracle Privileged Account Manager applies grants in the following order:

- User grants are given first priority.

  If a user has direct access to an account through a user grant, then Oracle Privileged Account Manager applies the Usage Policy that corresponds to that grant.

- If Oracle Privileged Account Manager cannot find a user grant for the user, then it looks for any group grants that grant the user access to that account.

  If the user is a member of multiple granted groups, then Oracle Privileged Account Manager sorts the group names into alphabetical order and uses the Usage Policy assigned to the first group.

  For example, assume you have Group A with corresponding policy *UsagePolicyB* and Group B with *UsagePolicyA*. When Oracle Privileged Account Manager sorts the group names, Group A comes first alphabetically, so Oracle Privileged Account Manager will apply UsagePolicyB.

10.3.1.2 Configuring Usage Policies for Users with Session Access

Oracle Privileged Session Manager supports SSH in both interactive (shell) mode and non-interactive (Exec) mode. Users can also copy files from a target by using Secure Copy (SCP).

You can configure the Usage Policy to constrain and enforce which tasks the privileged users are allowed to perform when they have session access. You can apply this control at the following levels:

- **Session mode level:** In this mode, you can control whether the user can use SCP or start an SSH session in Interactive or Non-Interactive modes.
  - **Interactive mode:** In this mode, the user can start a shell with the target.
  - **Non-Interactive mode:** In this mode, the user can execute a command remotely.
- **Command level:** In this mode, you can control which commands the user can execute on the target system in a SSH session.

  If you enable either interactive or non-interactive mode, then you can use Oracle Privileged Session Manager's command control and replacement feature.

Command Control

Using a whitelist or blacklist of commands, you can configure the commands that a user can or cannot execute on the target.

> **Note:** The commands in the list use java regular expression syntax.
>
> For example, if you specify `ls.*` all the commands that start with `ls` will be matched.

The following list describes how the whitelist and blacklist must be used:

- **Whitelist:** Use a whitelist to restrict the allowed commands to a defined set. This is the recommended option.
- **Blacklist:** Use a blacklist to prevent unintentional usage of commands that are deemed harmful. By using a blacklist and recording user activities, you can dissuade a user from executing such commands.

Command Replacement

You can specify a list of command names along with their replacements. This is useful to replace the execution of potentially harmful commands with their safer equivalents.

> **Note:** The commands in the command replacement do not support regular expression.

Only the command names are replaced, and the arguments are retained as is. For example, if the command "rm" must be replaced with "rm -i," then the sample input and executed command may be as follows:

Input command: `rm importantFile`

Executed command: `rm -i importantFile`

10.3.2 Searching for Usage Policies

Perform the following procedure to search for a Usage Policy:

1. Select **Usage Policies** from the Administration accordion.

2. When the Search Policies portlet is displayed, enter your search criteria into one or more of the following fields:
   - **Name**: Enter all or any part of a policy name.
   - **Status**: Select **All** (*default*) from the menu to search for all policies (active and inactive). Select **Active** or **Disabled** to limit the search to just active or inactive policies.
3. Click **Search**.

The search results are displayed in the Search Results table, which includes the Name and Status.

10.3.3 Viewing Usage Policies

Perform the following procedure to review parameter settings for a Usage Policy:

1. Select **Usage Policies** from the Administration accordion.
2. When the Policies page is displayed, click **Search**.

   The existing policies will be display in the Search Results table.
3. Use one of the following methods to open a policy:
   - Click the row number next to the name of the policy and then click the **Open** icon located above the Search Results table.
   - Click the policy name (an active link) in the Search Results table.

     For example, clicking the **Default Usage Policy** link opens the Usage Policy: Default Usage Policy page.

   The Usage Policy page contains four tabs:
   - **General Fields:** This tab contains parameters used to specify general information about the policy.
   - **Capabilities:** This tab contains parameters that are used to control which type of checkouts users can perform, to enable or disable session recording and to configure session access.
   - **Usage Rules:** This tab contains parameters that govern the time zone to be associated with checking out a privileged account, when the account can be checked out, and when the check out expires.
   - **Grantees:** This tab provides information about the grantees who are authorized to use that account.

10.3.4 Modifying the Default Usage Policy

After evaluating the Default Usage Policy, you may want to modify the settings to better suit your environment.

> **Note:** Oracle recommends that you make a back-up copy of the Default Usage Policy before making any changes. You can use the `export` command as described in Section A.10.1, "`export` Command."

To modify the Default Usage Policy, perform the following:

1. Select **Usage Policies** from the Administration accordion.

2. When the Usage Policies page is displayed, click **Search** to populate the Search Results table.
3. Select the **Default Usage Policy** link in the Search Results table to open the Usage Policy: Default Usage Policy page.
4. Select the General Fields tab, where you can modify the following parameter:

   Note: You cannot edit the **Name** or **Status** values for this policy.

   Description: Highlight and delete the existing text, and then enter your new description.
5. Select the Capabilities tab to do the following:
   - In the Basic Configuration area, you can modify any of the following parameters:
     – **Allow Checkout Type**: Use this menu to specify "All," "password," or "session" as the checkout option for this policy. The following are descriptions for these options:

       - All (*Default*): Specify this option to allow users to check out passwords and sessions.

       - password: Allow users to only check out passwords.

       - session: Allow users to only check out sessions.

       The following table lists the Session Checkout Settings that you can view or modify for the **All** or the **Session Checkout** option:

Parameter	Description
SCP Configuration	The Enable SCP (Secure Copy) setting displays a blue check mark if you have enabled SCP. Enabling this option enables the user to securely copy files to and from the target.

Parameter	Description
SSH Configuration	The Enable Interactive Mode and the Enable Non-Interactive Mode settings will display a blue check mark if they have been enabled. If either Interactive Mode or the Non-Interactive Mode is enabled, then in the **Command Control List** area you can select one of the following options for List Type: ▪ **None:** Command control is not applied in this case. ▪ **Black List:** Select this option to add a command to or remove a command from the black list. To add a command to this list, select **Black List** under List Type and click **Add** to insert a new row. In the new row, enter a command using java regular expression syntax in the Command column and click **Save.** To remove a command from this list, locate and select the command to be removed from the black list and click **Remove.** ▪ **White List:** Select this option to add a command to or remove a command from the white list. To add a command to this list, select **White List** under List Type and click **Add** to insert a new row. In the new row, enter a command using java regular expression syntax in the Command column and click **Add.** To remove a command from this list, locate and select the command to be removed from the white list and click **Remove.** The **Enable Command Logging** option will display a blue check mark if it has been enabled. Enable this flag to enable interactive session transcript. **Note:** Interactive session transcript is automatically available if the Command Control or Command Replacement option is enabled. In the **Command Replacement** area, you can specify the command to be replaced with its specified replacement. To do so, click **Add** to insert a new row. In the new row, enter the command name you want to replace in the Original column and enter the new command in the Replacement column and click **Add.** **Note:** The Command Replacement feature only replaces the command name, but retains the arguments. You cannot use regular expression in command replacement.

– **Enable Session Recording**: Select the enable session recording checkbox when this Usage Policy is applied to a session checkout.

Refer to Section 9.7, "Viewing a Session Recording" for more information about session recordings.

6. Select the Usage Rules tab to change one or more of following parameter settings:

Parameter	Description
Timezone	Select a time zone from the menu to indicate when the policy will be applied. For example, if you set the time zone to GMT, and the policy allows check-outs between 9am to 5pm, you can only check out between 9am-5pm GMT, and not PST.
Permitted Usage Dates	Use the **Monday** through **Sunday** checkboxes and the **From** and **To** drop menus to specify when grantees are allowed to use the account. Select one or more days of the week and the periods of time when grantees can access this account. The default access period is `24x7`.

Parameter	Description
Expiration	Enable one of the following options to change when the grantees' access to the account expires: ■ **Automatically check in account**. Use the counter to specify the number of **minutes after last check out**. ■ **Automatically check in account on this date**. Click the **Calendar** icon to open a Select Date and Time dialog. Use the month and year menus or click a day in the calendar to specify an expiration date. Use the hours, minutes, and seconds menus and enable the AM or PM buttons to specify an expiration time. ■ **Never expire**. No expiration period is required for the account. **Note**: The Oracle Privileged Account Manager scheduler periodically checks for accounts that have passed their specified expiration period and resets them as described in this section. The scheduler makes this check every 60 minutes by default (based on the `policyenforcerinterval` property in the OPAM Global Config configuration entry, whose default setting is *`60 minutes`*). You can view and modify the current interval by using Oracle Privileged Account Manager's `getconfig` and `modifyconfig` command line options. For more information, refer to Section A.2.1, "`getconfig` Command" and to Section A.2.3, "`modifyconfig` Command."

Note: If you are configuring a Usage Policy for a *shared* privileged account, it is prudent to configure an Automatic check-in option to ensure the account gets checked-in and the password gets cycled in a timely manner.

In addition, consider limiting how many users can access the shared account and further segregate these users by specifying when they can access the account. By specifying which days of the week and what times of the day each user can access the account, you minimize overlapping checkouts and improve Oracle Privileged Account Manager's auditing ability.

For more information about shared accounts, refer to Section 2.4.2, "Securing Shared Accounts."

---

7. Select the Grantees tab to view which grantee this policy is assigned to.

Note: To specify a different Usage Policy for any grantee listed in the table, click the name of the account which is an active link. When the Account page is displayed, select a different policy name from the Usage Policy menu.

Tip: Clicking the active links in the "Grantee Name" or "Account Name" columns enable you to navigate to other screens to see additional information.

8. When you are finished editing the policy, click **Apply** to save your changes.

10.3.5 Creating a Usage Policy

To create a Usage Policy,

1. Select **Usage Policies** from the Administration accordion.
2. When the Policies page is displayed, click **Create** at the top of the Search Results table.

   A new, "Usage Policy: Untitled" page is displayed with three tabs.
3. Provide the following information on the General tab:
   a. **Name**: Enter a name for the new policy.
   b. **Status**: Click the **Active** or **Disabled** button to specify whether the policy status is active or disabled

      Making the policy Active puts that policy into effect for the associated accounts and grants.

      Disabling a policy applies the Default Usage Policy to all accounts and grants associated with that disabled policy. If you simply assigned a different policy to those accounts and grants, you would lose all information about the old policy assignment.
   c. **Description** (*optional*): Enter a descriptive statement about the new policy.
4. Select the Usage Rules tab to define rules for using a privileged account. Refer to the table in Step 6 of Section 10.2.3, "Modifying the Default Password Policy" for a description of these parameter settings.
5. Select the Capabilities tab to control the checkout capabilities. Refer to the table in Step 5 of Section 10.3.4, "Modifying the Default Usage Policy" for a description of the Capabilities tab.
6. Select the Grantees tab to assign the new policy to accounts or grantees. Refer to Section 10.3.6, "Assigning Usage Policies" for detailed instructions.

   After assigning this policy, you can select the Grantees tab to review which users or groups are using this policy.
7. Click **Save**.

10.3.6 Assigning Usage Policies

When you create a new grant, Oracle Privileged Account Manager automatically assigns the Default Usage Policy to that grant. However, if you have created additional Usage Policies, as described in Section 10.3.5, "Creating a Usage Policy," then you can assign a different policy to the grant.

> **Note:** Administrators with the *User Manager* Admin Role can assign a Usage Policy to accounts at the *grantee-account pair* level. In other words, the User Manager can assign different Usage Policies to different grantees of the same account.

You can assign a different Usage Policy

- From the Accounts Page
- From the Targets Page
- From the Usage Policies Page

Note: When you add grantees to an account, as described in Section 11.2, "Granting Accounts to Users" or Section 11.3, "Granting Accounts to Groups," Oracle Privileged Account Manager adds the user or group name to the Users or Groups table on the Grants tab and automatically assigns the Default Usage Policy.

From the Accounts Page

To assign a Usage Policy from the Accounts page,

1. Locate the account where you want to assign the policy.
   a. Select **Accounts** in the Administration accordion.
   b. Click **Search** in the Search Accounts portlet to populate the Search Results table with a list of all available accounts.

      To narrow the results or to locate a particular account, enter search criteria in one or more the Search Accounts fields, and then click **Search**.
2. Locate the account's Account Name link to open the Account: *AccountName* page.
3. Select the Grants tab.
4. Locate the grantee in the Users or Groups table, and use the **Usage Policy** menu in that row to select a different policy.
5. Click **Apply** to add your changes.

From the Targets Page

To assign a Usage Policy from the Targets page,

1. Locate the target where the account is located.
   a. Select **Targets** in the Administration accordion.
   b. Click **Search** in the Search Targets portlet to populate the Search Results table with a list of all available targets.

      To narrow the results or to locate a particular target, enter search criteria in one or more the Search Targets fields, and then click **Search**.
2. Click the target name of the account (an active link) in the Search Results table to open that target.
3. When the "Target: *TargetName*" page is displayed, click the **Grants** tab to view a list of the grantees currently granted access to that account.

   Notice that the table lists the Usage Policy that is currently assigned to each grantee.
4. Locate the grantee in the Users or Groups table, and use the **Usage Policy** menu in that row to select a different policy.
5. Click **Apply** to finish assigning the policy to the selected account.

From the Usage Policies Page

To assign a Usage Policy from the Policies page,

1. Locate the Usage Policy that you want to assign to the account.
   a. Select **Usage Policies** in the Administration accordion.

b. Click **Search** in the Search Policies portlet to populate the Search Results table with a list of all available Usage Policies.

   To narrow the results or to locate a particular policy, enter search criteria in one or more the Search Policies fields, and then click **Search**.

2. When the search results display, locate the policy you want to assign. Click the Name link to open the Usage Policy: *PolicyName* page.
3. Select the **Grantees** tab.
4. Locate the user or group name in the Grantees table and then click the located account name (an active link) of the grantee to open the account.
5. When the "Account: *AccountName*" page is displayed, click the **Grants** tab.
6. Locate the grantee in the Users or Groups table, and use the Usage Policy menu in that row to select a different policy.
7. Click **Apply** to add your changes.

10.3.7 Deleting Usage Policies

Note: You cannot delete the Default Usage Policy.

To delete a Usage Policy,

1. Locate and select the policy to be deleted.
2. Click the **Delete** icon.
3. When the Confirm Remove dialog box is displayed, click the **Remove** button.

   The policy is immediately deleted. If you had any accounts assigned to that policy, they will all revert to using the Default Usage Policy.

11

Working with Grantees

This chapter describes the different tasks you can perform when working with grantees in Oracle Privileged Account Manager.

> **Note:** You must be an Oracle Privileged Account Manager administrator with the *User Manager* Admin
>
> Role to add, edit, or delete grantees.

This chapter includes the following sections:

- Section 11.1, "What Are Grantees?"
- Section 11.2, "Granting Accounts to Users"
- Section 11.3, "Granting Accounts to Groups"
- Section 11.4, "Searching for Grantees"
- Section 11.5, "Opening a Grantee"
- Section 11.6, "Removing Grantees from an Account"

> **Note:** You can also use Oracle Privileged Account Manager's command line tool or Oracle Privileged Account Manager's RESTful interface to perform many of the tasks described in this chapter.
>
> If you prefer using these interfaces instead of the Oracle Privileged Account Manager Console, refer to Appendix A, "Working with the Command Line Tool" or Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface" for instructions.

11.1 What Are Grantees?

Grantees are users or groups in the identity store that have been granted access to a privileged account managed by an Oracle Privileged Account Manager administrator. Users cannot check out a privileged account unless they have been granted access to that account.

Oracle Privileged Account Manager evaluates grants in the following sequence:

1. When a user tries to access and check out an account, Oracle Privileged Account Manager looks for a user grant for that user. If Oracle Privileged Account Manager finds a user grant, then the user is permitted to check out the account based on that grant and its associated Usage Policy.

2. If Oracle Privileged Account Manager does not find a user grant, it looks for group grants. A user can be a member of many groups. If Oracle Privileged Account Manager finds a group grant for any one of the user's groups, then the user is permitted to check out the account based on that group grant and its associated Usage Policy.
3. If the user is member of multiple groups, and more than one of those groups is available in group grants - then Oracle Privileged Account Manager can pick any one of the matching group grants at runtime. It is indeterministic to say exactly which matching group grant of the multiple ones Oracle Privileged Account Manager will pick at runtime.
4. If Oracle Privileged Account Manager cannot find a user grant or a group grant, then the user is denied access.

Note: Before granting privileged accounts to users or groups, be sure to read, Section 2.4.4, "Avoiding Assignments through Multiple Paths."

11.2 Granting Accounts to Users

Use the following steps to grant access to a privileged account:

1. Locate the account where you want to grant access.
   a. Select **Accounts** in the Administration accordion.
   b. Click **Search** in the Search Accounts portlet to populate the Search Results table with a list of all available accounts.

      To narrow the results or to locate a particular account, enter search criteria in one or more the Search Accounts fields, and then click **Search**.
2. Select that account name in the Search Results table.

   The "Account: *Account Name*" page is displayed with the General, Grants, Credential Store Framework, and Checkout History tabs.
3. Select the Grants tab.

   If any users are already associated with this account, their names are listed in the table in the Users area.
4. Click **Add** to open the Add Users dialog.
5. In the Add Users dialog, enter all or part of a user name and then click **Search**.

   For example, if you want to add the `jjones` user, then you could type **`j`**, **`jj`**, or **`jon`** and the search results will include any user names containing those letters.
6. Select (check) one or more user names, and then click **Add** to make them grantees.
7. Click **Close** to close the dialog.

   The new user's name is displayed in the Users table.

Note: At this point, the Default Usage Policy is automatically assigned to the user. However, you can use the Usage Policy menu to select a different policy for that user.

11.3 Granting Accounts to Groups

Use the following steps to grant access to a privileged account:

1. Locate the account where you want to grant access.
   a. Select **Accounts** in the Administration accordion.
   b. Click **Search** in the Search Accounts portlet to populate the Search Results table with a list of all available accounts.

      To narrow the results or to locate a particular account, enter search criteria in one or more the Search Accounts fields, and then click **Search**.
2. Select the account name in the Search Results table.

   The "Account: *Account Name"* page is displayed with the General, Grants, Credential Store Framework, and Checkout History tabs.
3. Select the Grants tab.

   If any groups are already associated with this account, their names are listed in the table in the Groups area.
4. Click **Add** to open the Add Groups dialog.
5. In the Add Groups dialog, enter all or part of a group name and then click **Search**.

   For example, if you want to add the `hr_admin` group, then you could type **`h`**, **`hr`**, or **`admin`** and the search results will include any group names containing those letters.
6. Select (check) one or more group names, and then click **Add** to make them grantees.
7. Click **Close** to close the dialog.

   The new group name is displayed in the Groups table.

Note: At this point, the Default Usage Policy is automatically assigned to the group. However, you can use the Usage Policy menu to select a different policy for that group.

11.4 Searching for Grantees

If you have administrator privileges, you can search for grantees by using the following steps

1. Select **User Grantees** or **Group Grantees** in the Administration accordion.
2. When the User Grantees or the Group Grantees page is displayed, use the Search portlet to configure your search.
   - To search for a particular grantee, enter one or more letters of the name into the **User Name** or **Group Name** field.
   - To search for all available grantees, do not specify any search parameters.
3. Click **Search**.

   Review your search results in the Search Results table.
4. To perform another search, click **Reset**.

11.5 Opening a Grantee

You can open a grantee to view information about that user or group grantee.

Use one of the following methods to open a grantee from the User Grantees or the Group Grantees page:

- Click the User Name or the Group Name (an active link) in the Search Results table.
- Select the user or group Row number and then click the **Open** icon.

The User: *UserName* or the Group: *GroupName* page opens where you can review the information about that grantee and the privileged accounts for which they are granted access.

11.6 Removing Grantees from an Account

Note: Removing a user or group grant from an account *does not* automatically cancel all existing check-outs.

When grantees check out an account, they are guaranteed access to that account until one of the following events occur:

- The user checks in the account
- Oracle Privileged Account Manager automatically checks in the account because the checkout duration has exceeded the expiration period specified by the account's Usage Policy
- An administrator forces an account check-in

However, after the account is checked in, the grantee cannot check out that account again unless an administrator re-adds them as a grantee.

To remove one or more grantees from an account

1. Open the account and select the Grants tab.
2. Select the user or group Row number in the Search Results table.
3. Click the **Remove** icon.
4. When you are prompted to confirm the removal, click the **Remove** button to continue, or the **Cancel** button to terminate the operation. Then the prompt closes and the user or group is removed from the table.

12

Working with Resource Groups

This chapter describes how to create, manage, and delegate access privileges for Oracle Privileged Account Manager resource groups in the following sections:

Note:

- You must be an administrator with the *Security Administrator* Admin Role to create or delete a resource group.
- You can also use the Oracle Privileged Account Manager's command line tool or RESTful interface to work with resource groups. For more information, refer to Section A.8, "Working with Resource Group" to use the command line tool and the Appendix B.11, "Resource Groups Resource" to use the RESTful interface.

12.1 What is a Resource Group?

In Oracle Privileged Account Manager, all targets and accounts are considered resources. A resource group is a collection of resources that can include targets, accounts, and other resource groups. Resource groups facilitate easier and better administration of resources in your deployment.

Resource groups simplify management by organizing data into groups and delegating administration to users or user groups. In Oracle Privileged Account Manager, a user with a global administrative role such as Security Administrator role has administrative access to all resources, such as, all targets and accounts. Deployment needs will require administrative access to be provided for users to a subset of resources rather than providing a global access. For example a regional admin may need access to manage only the resources within his region. Resource groups provide the mechanism to create such sub sets of resources and delegate administration to users.

For example, you could create a resource group called "Austin Group", that contains the two following members:

- Austin DB Group:

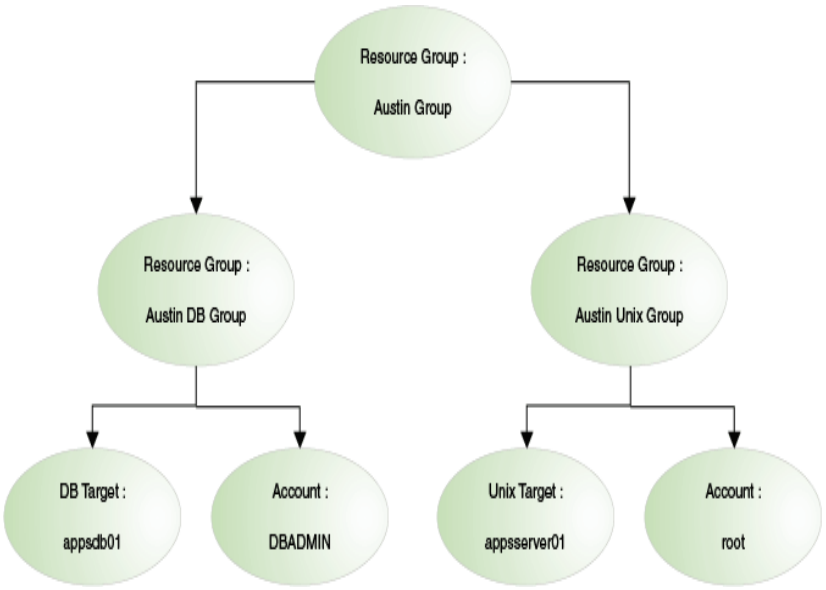
  This group further contains two member groups called "appsdb01" and "DBADMIN."

- Austin Unix group:

  This group further contains two member groups called "appserver01" and "root."

Figure 12–1 illustrates the structure of the "Austin Group" resource group.

Figure 12–1 Tree Structure of the "Austin Group" Resource Group



After creating a resource group, you can then delegate administration privileges for that group to specific users, groups, or both. For more information about delegation, refer to Section 12.3, "Delegating Administrative Privileges."

12.2 Creating Resource Groups

To create a resource group,

1. Log in to Oracle Privileged Account Manager.
2. Select **Resource Groups** from the Administration accordion. The Resource Groups page will open.
3. Click **Create** to open the "Resource Group: Untitled" page, which contains the following tabs:
   - **General:** Select this tab to create the resource group and add members.
   - **Delegate:** Select this tab to delegate and modify administrative privileges for the resource group to specific users and groups.
   - **Hierarchy View:** Select this tab to view and work with a hierarchical view of the selected resource group.
   - **Member-of:** Select this tab to view the different resource groups that this resource group is a member of, as resource groups can be hierarchical.

Note: The Delegate, Hierarchy View, and Member-of tabs do not become active until you create and save the resource group.

4. On the General tab, enter a name for the group in the "Name" field and add an optional description. For example, describe the purpose of this resource group to other administrators in the "Description" field.
5. In the Members section, click **Add**.

   The Add Members dialog box is displayed. Use this dialog box to search for and add members to the new resource group, as follows:

   a. Use the **Type** menu to indicate the type of members you are adding to the group. You can choose the **account**, **target**, or **resource group** options.

   b. Enter one or more letters of the name of an account, target, or resource group into the **Name** field, and click **Search**.

   c. After the search results are displayed, select one or more rows in the table, and click **Add** to add those members to the group.

      Tip: Use **Shift**+**Click** to select multiple, consecutive rows or **Ctrl**+**click** to select multiple, non-consecutive rows.

      A success message is displayed below the **Name** field.

   d. When you are finished, click **Close**.

      The selected members are displayed in the Members search results table.
6. Click **Save**.

Oracle Privileged Account Manager assigns a unique GUID value to the new group and displays that value under the **Name** field.

The Delegate tab becomes active. To delegate administrator privileges for the new resource group, refer to Section 12.3, "Delegating Administrative Privileges."

12.3 Delegating Administrative Privileges

This section explains delegation and describes how to delegate and remove access privileges for an Oracle Privileged Account Manager resource group. The following sections discusses this topic:

12.3.1 Understanding Delegation

Delegation enables you to share administrative responsibilities for a particular resource group or individual resources with other users or groups.

If an administrator delegates the administrative privileges over a resource group to a user or a group, then the delegatee only has administrative privileges for the designated resource. After delegation, the delegated user has privileges on the

resource group and all the resources within that resource group. The delegatee does not have privileges for all other resources outside of the delegated resource group.

Users who have delegated privileges can also delegate their delegated resources to other users or groups. However, only those privileges that were delegated to the user can be further delegated by the user. For example, a user with the "security administrator" privileges can only delegate "security administrator" privileges to others. Similarly, a user with the "user manager" privileges can only delegate the "user management" privileges to others.

For example, Jane is a user with the "Security Admin" privilege on the system and during her vacation, an LDAP server may need to be managed. In such a scenario, Jane can create a resource group and add the target to this group. She can delegate her "Security Admin" privilege for this resource group to Joe. Now, Joe will have the "Security Admin" privilege for Jane's resource group, and will be able to administer the group. However, he does not have the same privileges on any other resource.

Delegating privileges provides the following benefits:

- Flexibility.
- Makes it easier to separate roles and responsibilities in the system.
- Makes auditing safer and easier.

The following privileges can be delegated:

- Security Administrator
- User Manager

Initially, you may have users with Security Administrator or User Manager role for the entire system, and they can create resource groups and assign resource group level delegated administrators. Delegated administrators can further delegate to their privileges to other users and can also create sub resource groups to split the resource management. The following table explains the tasks that can be performed by the global administrators and delegated administrators:

Table 12–1 Tasks that each Admin Role can perform

Admin Role	Task Description
Security Administrator	■ Create, search for, view, modify, and delete resource groups. ■ Create new resources. ■ Add resources to, view resources, and remove resources from a resource group. ■ Delegate their security administration privilege for a resource group to other users and groups.

Table 12–1 (Cont.) Tasks that each Admin Role can perform

Admin Role	Task Description
User who is delegated the Security Administrator privilege for a resource group	■ Create resource groups under the delegated resource group. Search for, view, modify and delete delegated resource groups. ■ Add resources to, view resources and remove resources from a delegated resource group. ■ Create and add new resources to a delegated resource group. ■ Further delegate their administration privileges for a delegated resource group to other users and groups.
User Manager	■ Search for and view resource groups. ■ View member resources of a resource group. ■ Search for, view, or modify the User Management privilege of a resource group to other users or groups.
User or group to whom the User Manager privilege for a resource group is delegated	■ Search for and view delegated resource groups. ■ View member resources of delegated resource groups. ■ Search for, view, or modify the User Management privilege of delegated resource groups to other users or groups. ■ Further delegate their administration privileges for a delegated resource group to other users and groups.

Note: When you delegate privileges on a resource group, you are implicitly granting the same administration privileges for all resources in that group, including resource groups that are part (child groups or children) of the delegated resource group.

12.3.2 Delegating Privileges to Users and Groups

To delegate access privileges for a resource group to users or groups, select the Delegate tab and complete the following steps:

Note: The steps for adding and delegating privileges to users and groups are essentially the same.

- To add users, perform these steps from the Users section.
- To add groups, perform these steps from the Groups section.

1. Click the **Add** icon located above the Search Results table.
2. When the Add Users dialog box is displayed, search for available delegates by typing one or more letters of a name into the **Name** field. Click **Search**.

   For example, if you enter a single character that is common to many names, a list of all names containing that character will display. To narrow the results, enter more characters.
3. Select one or more rows from the list of results and click **Add**.

   Tip: Use **Shift**+**Click** to select multiple, consecutive rows or **Ctrl**+**click** to select multiple, non-consecutive rows.
4. After selecting the desired rows, Click **Close**.

The selected names and the assigned Privilege are now available in the appropriate table on the Delegate tab.

5. To change the default assigned privilege, choose a different option from the **Privilege** menu.
6. When you have finished adding delegates, click **Apply** to save your changes.

12.3.3 Opening Users and Groups

You can open a "delegate" to view more information about that user or group as described below:

For users you can view the following information:

- First Name and Last Name
- User Type
- OPAM Role
- The privileged accounts that the user can access

For groups you can view the following information:

- Group description
- List of users who belong to the group
- List of groups that belong to this group
- The privileged accounts that the group can access

Perform the following procedure to open a delegate:

1. Select the Name link in the Users or Groups table.
2. Select the appropriate table row and click **Open**.

Depending on your selection, the User: *User_Name* page or the Group: *Group_Name* page is displayed.

12.3.4 Removing Access Privileges

To remove a delegate and their access privilege, select the table row in the Users or Groups table and click **Remove**.

Note: The user who was delegated a resource group can modify delegations on this resource group. The delegated user can even remove the delegated privilege. Do *not* remove the delegated privilege, as the only workaround for a removed delegated privilege is to have it re-delegated.

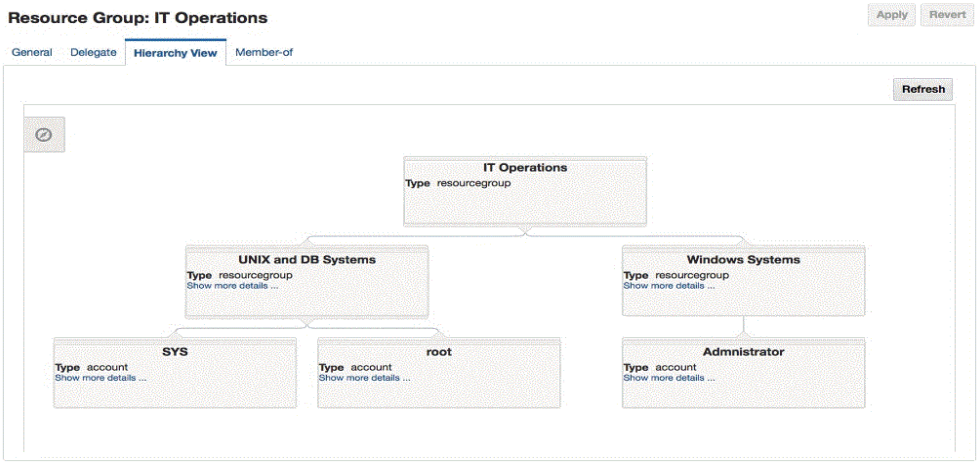
12.4 Working with Hierarchical Views

This section describes how to view and work with the hierarchy of a resource group.

After you create and save a resource group, the Hierarchy View tab becomes active. Select this tab to view a diagram of the resource group's hierarchy, which includes the parent resource group and all of the resource members (accounts, targets, and other resource groups).

For example, Figure 12–2 shows a diagram of a resource group named IT Operations.

Figure 12–2 Example Hierarchical View of the `IT Operations` Resource Group



In addition to the diagram, this page also contains a control panel that enables you manipulate the hierarchical diagram in various ways. This control panel is hidden by default. To access the controls, click the **Show/Hide Control Panel** icon , located on the left side of the display area.

The control panel is displayed, as shown Figure 12–3:

Figure 12–3 Hierarchical Diagram Control Panel

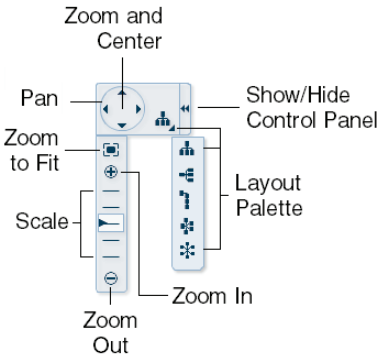


Table 12–2 describes the features that are available on this control panel.

Table 12–2 Hierarchy View Page Control Panel Features

Feature	Description
Zoom and Center	Click the center of the wheel to center the diagram within the display area.
Pan	Click the arrow points along the outside of the wheel to move the diagram around the display area.
Zoom to Fit	Click this icon to scale the diagram so that it fits into the display area.
Scale	Slide the arrow up or down the bars to scale the diagram size. Slide up to enlarge the diagram or down the scale to shrink it.

Table 12–2 (Cont.) Hierarchy View Page Control Panel Features

Feature	Description
Zoom Out	Click the icon to zoom out the diagram.
Zoom In	Click the icon to zoom in the diagram.
Layout Palette	Click the icon to display a palette of icons that you can use to change the type of diagram being displayed. The layout options include: ■ Vertical, Top Down ■ Horizontal, Left-to-Right ■ Tree ■ Layout ■ Circle

In addition to the control panel, the Hierarchy View page provides the following features:

- **Click and drag:** Click anywhere on the diagram and use your mouse to drag the diagram around the display area.
- **Rollover pop-up:** Hover your cursor over the following features to view more information:
  - Hover over icons in the Control Panel to view a pop-up containing the name and, if available, the alternative quick keys.
  - Hover over any box in the diagram to view a pop-up containing an enlarged view of the information in that box.
- **Show more details links:** Click a link to view the configuration page for that particular member. For example, if you click the link in a target box, the Target:*Target_Name* page is displayed.
- **Hide, Show, Isolate, and Restore icons:** Toggle the icon to collapse the diagram to view only the selected box and then restore the diagram to view all of the boxes.
- **Quick-Key commands:** Type the following key combinations as an alternative to using the control panel icons:
  - **`Ctrl+Alt+0`**: Zoom and Center
  - **`/`** (backslash): Toggle to hide or show the Hierarchical Diagram Control Panel
  - **`0`** (zero): Zoom to Fit
  - **`+`** (plus sign): Zoom In
  - **`-`** (minus sign): Zoom Out
  - **`Shift+Enter`**: Hide, Show, Isolate, and Restore

12.5 Searching for Resource Groups

If you have administrator privileges, you can search for a resource group by Name, Description, or by using both these parameters.

To search for a resource group,

1. Select **Resource Groups** in the Administration accordion.
2. When the Resource Groups tab is displayed, use the Search portlet parameters to configure your search. For example,

- To search for the `rg1` resource group, you could type **`r`**, **`rg`**, or **`rg1`** into the Name field.
- To search for all existing resource groups, do not specify any search parameters.

3. Click **Search**.

   Review your search results in the Search Results table.

12.6 Opening Resource Groups

You can open a resource group to review and edit the group's configuration parameters.

To open a resource group:

1. Select **Resource Groups** in the Administration accordion.
2. When the Resource Groups tab is displayed, use the Search portlet to locate the resource group you want to open.
3. When the results are displayed in the Search Results table, perform the following:
   - Click the resource group's Name (an active link) in the Search Results table.
   - Select the resource group's Row number and then click the **Open** icon.

The Resource Group: *ResourceGroupName* page opens where you can access the group's configuration and delegation information.

12.7 Deleting Resource Groups

> **Note:** Only administrators with the *Security Administrator* Admin Role can delete resource groups.

To delete a resource group,

1. Select **Resource Groups** in the Administration accordion.
2. When the Resource Groups tab is displayed, use the Search portlet to locate the group you want to delete.
3. Select the "row number" of the group from the Search Results table and then click the **Remove** icon.

13

Working with Plug-Ins

This chapter provides some background information about Java plug-ins for Oracle Privileged Account Manager and explains how to configure and deploy plug-ins by using the Oracle Privileged Account Manager Console.

This chapter includes the following sections:

Note: You can also manage Oracle Privileged Account Manager plug-ins from the command line or by using Oracle Privileged Account Manager's RESTful interface.

- For information about using the Oracle Privileged Account Manager Command Line Tool (CLI), refer to Section A.9, "Working with Plug-Ins" in Appendix A, "Working with the Command Line Tool."
- For information about using the Oracle Privileged Account Manager RESTful interface, refer to Section B.7, "Account Resource" in Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface."

13.1 What is a Plug-In?

A plug-in is a customized program that enables you to extend Oracle Privileged Account Manager's functionality to better meet your specific business and technical requirements. A plug-in enables you to provide custom logic as part of a transaction or by connecting to a custom data source.

An Oracle Privileged Account Manager plug-in can be a Java program that has a configuration entry in the Oracle Privileged Account Manager server. That configuration entry specifies the conditions for invoking the plug-in, which include:

- An operation, such as `checkout` or `update`
- A resource, such as an account, a target, or a server

- A timing, relative to the operation, such as `pre_checkout` or `post_update`

Oracle Privileged Account Manager plug-ins can provide various types of added functionality, such as:

- Validating data before the Oracle Privileged Account Manager server performs an operation on it and performing specified actions after the server performs an operation
- Sending notifications based on Oracle Privileged Account Manager operations
- Performing step-up authentication and authorization
- Authenticating users through external identity stores

Upon start-up, the Oracle Privileged Account Manager server loads your plug-in configuration and library. When the server processes requests, it calls the plug-in functions whenever the specified event takes place.

13.2 Developing Plug-Ins for Oracle Privileged Account Manager

This section provides an overview of how you develop plug-ins for Oracle Privileged Account Manager. The topics include:

13.2.1 Overview

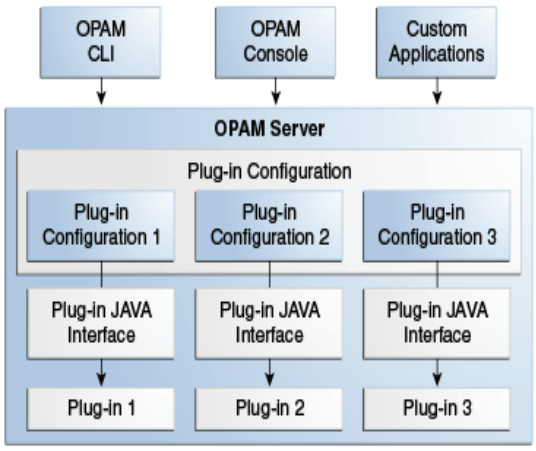
You can develop plug-ins by using the Oracle Privileged Account Manager plug-in framework, which is shipped in the following jar file:

ORACLE_HOME`/opam/jlib/opam-plugin-framework.jar`

Note: Currently, Oracle Privileged Account Manager does not ship with any complete plug-ins. Refer to Chapter 18, "Developing Plug-Ins for Oracle Privileged Account Manager" for additional information about developing plug-ins for Oracle Privileged Account Manager.

Figure 10-1 illustrates the Oracle Privileged Account Manager plug-in framework.

Figure 13–1 Oracle Privileged Account Manager Plug-In Framework



You can configure plug-ins for Oracle Privileged Account Manager operations. The plug-ins are invoked whenever the operations are performed and the plug-in filter rules are met. Any type of Oracle Privileged Account Manager client, such as the command line tool or the Oracle Privileged Account Manager Console, can perform these operations. The Oracle Privileged Account Manager server and the plug-in program communicate through the plug-in JAVA interface. When a plug-in is invoked, the Oracle Privileged Account Manager server sends information about the operation and the entities involved to the plug-in. The plug-in then operates on that information and, after completing execution, sends the result back to the Oracle Privileged Account Manager server.

Note: For additional information, refer to Section 18.3.1, "Communication between the Server and Plug-In."

After developing a plug-in, you register it with Oracle Privileged Account Manager. Registration enables Oracle Privileged Account Manager to discover and configure the plug-in to be invoked for Oracle Privileged Account Manager events such as check-ins, checkouts, and so forth.

Oracle Privileged Account Manager's Java-based plug-in framework enables you to create new plug-ins as well as customize existing ones.

13.2.2 Supported Languages

Currently, Oracle Privileged Account Manager only supports plug-ins written in Java.

13.2.3 Prerequisites

To develop Oracle Privileged Account Manager plug-ins, you should be familiar with the following topics:

- Oracle Privileged Account Manager
- Oracle Privileged Account Manager RESTful API

In addition, you should have some proficiency programming in Java.

13.2.4 Oracle Privileged Account Manager Plug-In Benefits

Some of the ways in which you can use plug-ins to extend Oracle Privileged Account Manager operations include:

- Validating data before the server performs operations on that data
- Performing actions that you define after the server successfully completes an operation
- Defining extended operations
- Authenticating users through external credential stores
- Replacing an existing server module with a module of your own

Upon start-up, the Oracle Privileged Account Manager server loads your plug-in configuration and library. The server calls your plug-in functions while processing various Oracle Privileged Account Manager requests.

13.2.5 Design Guidelines

Oracle recommends using these guidelines when designing plug-ins:

- Use plug-ins to guarantee that when Oracle Privileged Account Manager performs a specific operation, related actions are also performed.
- Use plug-ins only for centralized, global operations that should be invoked for the program body statement, regardless of which user or application issues the statement.
- Do not create recursive plug-ins.

  For example, creating a `pre_checkout` plug-in that itself issues a `checkout` statement will cause the plug-in to execute recursively until it has run out of resources.
- Use plug-ins judiciously. Remember, they are executed every time the associated operation occurs.

13.2.6 Framework Description

The Oracle Privileged Account Manager plug-in framework is the environment in which you develop, configure, and apply plug-ins. Each individual plug-in instance is called a *plug-in module*.

The Oracle Privileged Account Manager plug-in framework includes the following:

- Plug-in configuration tools
- Plug-in module interface
- Plug-in RESTful APIs, including Java package oracle.*xxx*.*xxx*

To use the Oracle Privileged Account Manager server plug-in framework,

1. Write a user-defined plug-in procedure in Java.
2. Compile the plug-in module.
3. Register the plug-in module through the configuration entry interface by using Oracle Privileged Account Manager's
   - Console, as described in Section 13.3, "Creating a Plug-In Configuration."
   - Command line tool, as described in Section A.9, "Working with Plug-Ins."

- RESTful interface, as described in Section B.12, "Plug-In Resource."

13.2.7 Supported Operations and Timings

Oracle Privileged Account Manager supports plug-ins for operations on the following resources:

Operations For Account Resources	Operations For Server Resources	Operations For Target Resources
■ add	■ accountpasswordchange (post)	■ add
■ checkin	■ autocheckin (pre/post)	■ remove
■ checkout	■ passwordcycle (pre/post)	■ resetpassword
■ remove		■ retrieve
■ resetpassword		■ showpassword
■ retrieve		■ test
■ sessioncheckout		■ update
■ showpassword		
■ showpasswordhistory		
■ test		
■ update		

When developing plug-ins, Oracle Privileged Account Manager enables you to specify when those plug-ins are executed. Oracle Privileged Account Manager supports *pre* and *post* operation timings for plug-ins, which are described in the following sections:

- Section 13.2.7.1, "Pre-Operation Plug-Ins"
- Section 13.2.7.2, "Post-Operation Plug-Ins"

13.2.7.1 Pre-Operation Plug-Ins

Oracle Privileged Account Manager adds pre-operation plug-ins to a queue and executes them in a specified order before performing the designated operation. As Oracle Privileged Account Manager executes each pre plug-in that is in the queue, the results are passed from the current plug-in to the next one in the queue. For example, if a target-add pre plug-in modifies the plug-in description, then the next plug-in sees the modified description.

Oracle Privileged Account Manager will not perform the operation until all pre plug-ins have successfully finished executing. If a pre plug-in fails or times out, then the operation also fails and Oracle Privileged Account Manager will not execute any of the other pre plug-ins queued for that operation.

Adding lots of pre plug-ins may increase the time taken for the operation because Oracle Privileged Account Manager must execute all pre plug-ins before performing the operation.

13.2.7.2 Post-Operation Plug-Ins

Oracle Privileged Account Manager adds post-operation plug-ins to a queue and executes them after performing the designated operation. Oracle Privileged Account Manager executes post plug-ins for both successful and failed operations. Post plug-ins assess the results, including the success or failure status of the operation, but they cannot modify the results because the operation has already been completed. Oracle Privileged Account Manager executes all queued post plug-ins and it does not

matter if some of them fail. Results from one post plug-in cannot be passed to another. The execution order of post plug-ins is arbitrary and is not guaranteed and the 'order' attribute is not effective here.

Adding post plug-ins does not increase the time taken for the operation because they are not executed until after the operation is performed.

Retrying Post Plug-ins

Oracle Privileged Account Manager post-operation plug-ins can be used to perform tasks like synchronizing password changes to external application wallets. There can be cases when the post plug-in may fail due to non-product issues such as network unavailability, remote host down, and so on. In such cases, the post plug-in can be retried to ensure that the extended functionality is completed when the condition for failure is fixed.

The Oracle Privileged Account Manager plug-in framework supports a retry mechanism and allows administrators to configure the retry parameters.

The Oracle Privileged Account Manager post plug-in configuration will queue the failed plug-in execution and retry them periodically when the retry parameters are defined. The configuration parameters control the duration of retry for the failed plug-in. It also controls when the retry is abandoned.

The following are the parameters of the retry configuration:

Parameter	Description
Retry duration in seconds	This is the duration (specified in seconds) of the interval or the waiting period before next re-run of the plug-in.
Retry number of times	This is the total number of times the plug-in must re-run before abandoning the retry.

13.2.8 Filtering Rules

The Oracle Privileged Account Manager server executes plug-ins on specifically configured operations for specific end users. You can configure filtering rules that determine for which users or groups a plug-in will or will not be executed, and for which results codes a post plug-in will be executed.

Note: Only administrators with the *Security Administrator* Admin role can add, edit, or remove Filter Rules.

- To configure filtering rules from the Console, refer to Section 13.3, "Creating a Plug-In Configuration," step 6 for instructions.
- To configure filtering rules from the command line, refer to Section A.9.1, "`addplugin` Command," for instructions.

Oracle Privileged Account Manager evaluates filtering rules in the following sequence to decide which rule takes precedence over another:

1. Enable Result Code (for post plug-ins)
2. Disable Result Code (for post plug-ins)
3. Disable user
4. Enable user

5. Disable group
6. Enable group
7. Enable Resource Group
8. Disable Resource Group

In general, rules that are defined at the user level override those that are defined at the group level, because the user level rules are more specific. User level rules target a specific user rather than a group of users.

In addition, the Disabled lists take precedence over Enabled lists.

Note: Result codes override all other filtering rules.

For Users and Groups

You can specify the users or groups for which a plug-in should be executed by adding them to an Enabled user or group list. Similarly, you can specify the users or groups for which a plug-in should not be executed by adding them to a Disabled user or group list.

For example, assume *person1* is a user in the *Administrators* group. If you put person1 in the Disabled user list and Administrators into the Enabled group list, then the Oracle Privileged Account Manager will not invoke the plug-in for person1 because the server checks the Disabled user list before checking the Enabled group list, and because the user-level rule overrides the group-level rule.

However, if you put person1 in the Enabled user list and put Administrators in the Disabled group list, then person1 can invoke the plug-in because the Enabled user check is performed before the Disabled group check.

If there are no values in these four fields, then all users and groups can invoke the plug-in. However, as long as there is one user or group in the Enabled user list or Enabled group list, then only that user or group can invoke the plug-in. No others can invoke that plug-in. If person1 is the only user in the Enabled user list, then all other users and groups are prevented from invoking the plug-in.

Note: The Filtering Rules evaluation sequence *stops* when it finds a match. For example, if the filter finds an Enabled user that matches the user who is performing the action, then the filtering stage stops. It does not matter if the user is present in any Enable or Disable group filters.

For Results Codes

After performing an Oracle Privileged Account Manager operation, the server returns an HTTP response containing an HTTP status integer, such as 200 for success, 201 for creation, 401 for insufficient privileges, and so on.

You can configure filtering rules for post plug-ins that are based on one or more HTTP result code values. For example, if you specify a filtering rule for enabling result code 200, then the server will only execute the post plug-in when the result status is 200. And, if you specify a filtering rule for disabling result code 200, then the server will only execute the post plug-in when the result status is not 200.

Note: Result codes override all other filtering rules.

For Resource Groups

Accounts and Targets can be members of Resource Groups. You can configure filtering rules based on the resource groups to specify for which accounts and targets the plug-in will execute.

For example, assume that the structure of your deployment contains the "USAccounts" resource group holding accounts located in USA as members, and the "UKAccounts" resource group holding accounts located in UK as members. Here, some additional validations and notifications may be needed for accounts located only in the USA. In this case, the filtering rule can be established to enable the plug-in to perform this additional task for the "USAccounts" resource group. Once the filtering rule is established, whenever an account is checked out, Oracle Privileged Account Manager checks if the account is a member of the "USAccounts" resource group, and if yes, executes the plug-in.

13.3 Creating a Plug-In Configuration

Creating a plug-in configuration means to register details about the plug-in with the plug-in resource (an account, a server, or a target).

> **Note:** You must be an Oracle Privileged Account Manager administrator with the *Application Configurator* Admin Role to create plug-ins.
>
> When you create a new plug-in configuration, the status is disabled and the plug-in cannot be executed. Only an administrator with the *Security Administrator* Admin Role can enable plug-in configurations and determine under which conditions those plug-ins can be executed.

To create a plug-in configuration from the Console,

1. Select **Plug-in Configuration** in the Configuration accordion to open the Search Plug-in Configuration page.
2. Click **Create**, located in the Search Results table toolbar.
3. When the "Plug-In Configuration: Untitled" page is displayed, provide the following information in the Configuration Settings section of the General tab:

Parameter Name	Description
Name	Enter a name for the new plug-in configuration.
Description	Enter a description for this plug-in configuration.
Status	Enable an option to configure the plug-in's execution status at runtime by selecting one of these options: ■ **Active**: Allow the plug-in to execute. ■ **Disabled** (*default*): Do not allow the plug-in to execute.
Resource Type	Choose the type of resource on which the plug-in will perform: ■ **account** ■ **server** ■ **target**

Parameter Name	Description
Operation	Choose the operation that the plug-in will perform. **Note**: Refer to Section 13.2.7, "Supported Operations and Timings" for a complete list of supported operations.
Timing	Specify when you want Oracle Privileged Account Manager to execute the plug-in by choosing one of the following options: ▪ **pre**: Executes the plug-in *before* performing the Oracle Privileged Account Manager operation. ▪ **post**: Executes the plug-in *after* performing the Oracle Privileged Account Manager operation. **Note**: Refer to Section 13.2.7, "Supported Operations and Timings" for more information.
Order	Enter a value to specify the order in which a plug-in is queued for execution in relation to other plug-ins. For example, plug-in 1 is executed before plug-in 2. The minimum value is `1`.
Timeout	Specify a value to indicate the maximum duration (in seconds) for which the plug-in can be executed. When the plug-in execution exceeds this time out period, Oracle Privileged Account Manager aborts the plug-in execution. The default is *`120 seconds`*, and the minimum value is `10 seconds`. **Note**: For additional information, refer to Section 13.2.7, "Supported Operations and Timings."
Plug-in Class Name	Enter the name of the Java class that implements the plug-in's interface.
Plug-in Version	Enter the plug-in's Java version number. **Note**: Oracle Privileged Account Manager does not actually use the plug-in version. Instead, Oracle Privileged Account Manager uses the jar file listed in the plug-in's directory.
Retry Times	Specify the number of times the plug-in must re-run before abandoning the retry. **Note:** This can parameter be configured only for a post plug-in.
Retry Interval	Specify the duration (in seconds) of the interval or the waiting period before the plug-in is retried is it failed. **Note:** This can parameter be configured only for a post plug-in.

4. To configure a Java classpath where the plug-in jar file is located, use the Class Paths section as follows:

   – To add a classpath, click **Add**.

   When a new row is displayed in the table, type the Java classpath into the blank field. For example,

   ```
   /u01/plugins/emailplugin.jar
   ```

   Note: The Oracle Privileged Account Manager server process must be able to access the specified class path files. You can specify any type of location, such as local file system, network file system, etc.

   – To delete a classpath, select that classpath row in the table, and then click **Remove**.

5. To configure custom attributes for the plug-in, expand and use the Custom Attributes section as follows:

   – To add an attribute, click **Add**.

When a new row is displayed in the table, type the **Attribute Name** and **Value** into the blank fields. For example, for an email notification plug-in, you might create a **`notificationemail`** attribute with a value of **`abc@abc.com`**.

– To delete a custom attribute, select that attribute's row in the table, and then click **Remove**.

– To load custom attributes, if they are defined for a plug-in, click **Load.**

6. To configure filtering rules that determine when Oracle Privileged Account Manager executes the plug-in, select and use the Filter Rules tab as follows:

Note: You must be an Oracle Privileged Account Manager administrator with the *Security Administrator* Admin Role to add, edit, or remove Filter Rules.

For information about how Oracle Privileged Account Manager uses filtering rules, refer to Section 13.2.8, "Filtering Rules."

- Expand the Users, Groups, Resource Groups, or Result Codes sections to specify for which users or groups the server can or cannot invoke the plug-in.

  a. Select the Enabled tab or Disabled tab and click **Add**.

  b. When the Add dialog box is displayed, enter one or more letters of a name into the **User Name** field for Users, or into the **Name** field for Groups and Resource groups and click **Search**.

  c. When the search results display, select the row you want, and click **Add**.

     A success message is displayed above the search results list.

  d. Continue selecting and adding users or groups until you are finished, then click **Close**.

- Expand the Result Codes section to configure filtering rules for a post plug-in that are based on one or more result codes.

Note: You cannot configure result codes for *pre* plug-ins.

---

  a. Click **Add**.

  b. When a new **HTTP Result Code** row is displayed, enter an enabled HTTP response code into the blank field. For example, type **`200`** to execute a post plug-in when the response status is a successful request.

  c. When a new **HTTP Result Code** row is displayed, enter a disabled HTTP response code into the blank field to support cases such as to execute a post plug-in for failures only. For example, type **`200`** to execute all post plug-ins whose response status is not **`200`**.

- To delete a Filter Rule, select that rule row in the applicable table, and then click **Remove**.

7. After setting all of the necessary plug-in configuration parameters, click **Test** to verify that the configuration is valid.

   This test checks whether Oracle Privileged Account Manager can load the configured plug-in and whether it implements the required plug-in interface. Testing catches common issues, such as plug-in jars that are configured with the

wrong file paths, plug-ins that implement the wrong interface, pre plug-ins that implement a post plug-in interface or vice versa, etc.

This test does not execute the plug-in or validate any plug-in custom attributes, which are only used by the custom plug-in logic itself.

If the configuration is valid, a "`Test Succeeded`" message is displayed.

8. Click **Save** to create the new configuration.

   Oracle Privileged Account Manager automatically assigns a Plug-In GUID, which is displayed in the Configuration Settings section.

13.3.1 Creating a Duplicate Plug-in Configuration Using the Create-Like functionality

Administrators creating a new plug-in configuration may want to configure similar or duplicate plug-ins for multiple operations, such as, creating an email notification plug-in for both account checkout and checkin. In such cases, the Create Like functionality can be used, where a new plug-in configuration is created as duplicate of an existing configuration. Then, the administrator can edit only the required values and save the configuration without the need to reconfigure all the values in the Plug-in Configuration page. To do so:

1. Select **Plug-in Configuration** in the Configuration accordion. In the Plug-in Configuration page, Click **Search.**

   Note: You can use the Search portlet parameters to configure your search. Refer to Section 13.4, "Searching for Plug-In Configurations" for detailed information.

2. In the Search Results table, select the row of the plug-in you want to copy and then click **Duplicate,** located in the Search Results table toolbar.

   A "Copy of *PLUG-IN_NAME*" page opens. Here, *PLUG-IN_NAME* is placeholder text, and the name of the plug-in you selected will appear instead.

3. In the "Copy of *PLUG-IN_NAME*" page, you must specify values for the Name, and Description fields. You can change the values for any of the other fields to customise the plug-in configuration for a new plug-in. For detailed information about plug-in configuration parameters, refer to Section 13.3, "Creating a Plug-In Configuration."

4. Click **Save** to create the new plug-in configuration.

13.4 Searching for Plug-In Configurations

You can search for plug-in configurations by using one or more of the following parameters:

- Name
- Description
- Resource Type (**All**, **account**, **server**, or **target**)
- Status (**All**, **Active**, or **Disabled**)
- Timing (**All**, **pre**, or **post**)

- Operation (**All**, **accountpasswordchange**, **add**, **autocheckin**, **checkin**, **checkout**, **passwordcycle**, **remove**, **resetpassword**, **retrieve**, **sessioncheckout**, **showpassword**, **showpasswordhistory**, **test**, or **update**)

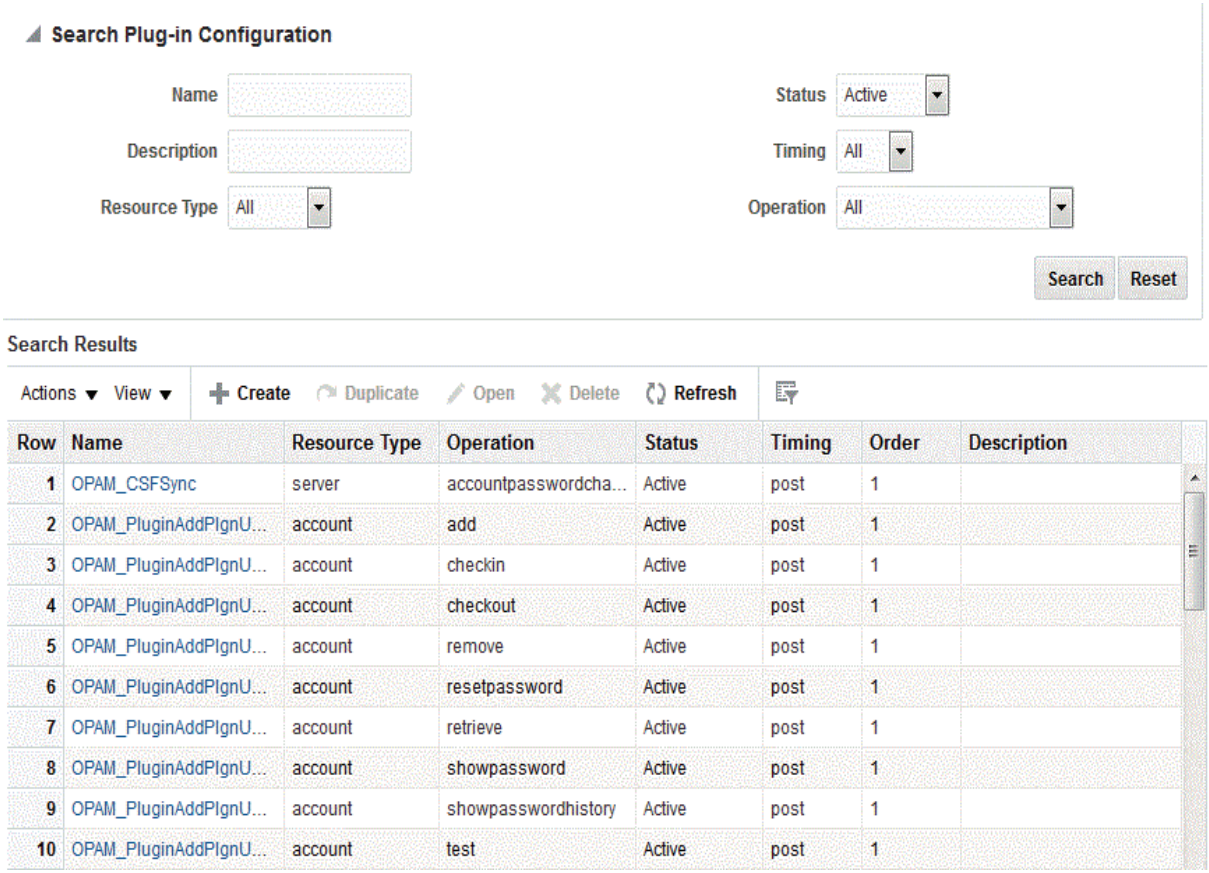
> **Note:** You must be an Oracle Privileged Account Manager administrator with the *Security Administrator* Admin Role or the *Application Configurator* Admin Role to search for and view plug-ins.

To search for a plug-in,

1. Select **Plug-in Configuration** in the Configuration accordion.
2. When the Plug-in Configuration page is displayed, use the Search portlet parameters to configure your search.
   - For example, to search for a list of all active plug-ins, select **Active** from the **Status** menu.
   - To search for all available plug-ins, do not specify any search parameters.
3. Click **Search**.

   Review your search results in the Search Results table, which contains a column for all of the search fields and a column for the Plug-In Order.

Figure 13–2 Example Plug-In Search Results



4. To perform another search, click **Reset**.

13.5 Opening a Plug-In

You can open a plug-in to view or edit the configuration parameters for that plug-in.

> **Note:** You must be an Oracle Privileged Account Manager administrator with the *Security Administrator* Admin Role or the *Application Configurator* Admin Role to view plug-ins.

To open a plug-in, open the Plug-in Configuration page and perform one of the following actions:

- Click the Plug-in Name (an active link) in the Search Results table.
- Select the plug-in Row and then click **Open**.

The Plug-in Configuration: *Plug-in Name* page opens where you can access the plug-in's configuration settings, custom attributes, users or groups, and current status (active or disabled).

If you edited any of these settings, click **Test** to validate your changes. If the test is successful, a "`Test Succeeded`" message is displayed. Click **Save**.

13.6 Deleting a Plug-In

To delete a plug-in configuration,

1. Locate the plug-in to remove.
   a. Select **Plug-in Configuration** in the Configuration accordion.
   b. Click **Search** in the Search portlet to populate the Search Results table with a list of all available plug-ins.

      To narrow the results or to locate a particular plug-in, enter search criteria in one or more the Search fields, and then click **Search**.
2. In the Search Results table, select the plug-in to be removed and then click **Delete**.

   When you are prompted to confirm the deletion, click **Delete** to continue or **Cancel**.

14

Working with Self-Service

This chapter provides instructions for self-service end users working with Oracle Privileged Account Manager.

This chapter includes the following sections:

14.1 Introduction to Using Self Service

Self-service users do not have any Oracle Privileged Account Manager administrator privileges or Admin Roles.

The basic workflow for a self-service user includes the following:

1. Viewing your accounts
2. Searching for accounts
3. Checking out accounts
4. Viewing checked-out accounts
5. Checking in accounts
6. Checking out a session
7. Viewing checked-out sessions
8. Checking in a session
9. Viewing an account password

Note: You can also use Oracle Privileged Account Manager's command line tool or Oracle Privileged Account Manager's RESTful interface to perform these tasks.

If you prefer using these interfaces instead of the Console, refer to Appendix A, "Working with the Command Line Tool" or Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface" for instructions.

14.2 Viewing Your Accounts

To view a list of all the accounts for which you are currently a grantee, select **My Accounts** on the Home accordion and then click **Search**.

The My Accounts page is refreshed and lists all of your accounts in the Search Results table. From this page you can

- Search your accounts.
- View the account name, the associated target name and target type, the domain, and a description.
- Open an account to review the associated target information and account information, which includes the Usage Policy associated with that account.
- Check out passwords and sessions.
- Control how information is displayed in the table by managing which columns are displayed and in which order.
- Refresh the list of displayed accounts after making changes.

14.3 Searching for Accounts

To search for an account, follow the instructions provided in Section 9.3, "Searching for Privileged Accounts."

14.4 Opening Accounts

To view information about an account for which you are a grantee:

You can open privileged accounts from the My Accounts or My Checkouts page.

From the My Accounts Page

1. Select **My Accounts** in the Home accordion.
2. Click **Search** to see all of your accounts displayed.

   Alternatively, you can narrow the results by configuring one or more of the Search Accounts parameters, as described in Section 9.3, "Searching for Privileged Accounts," and then click **Search**.
3. When the results display in the Search Results table, locate the account you want to open, and perform one of the following actions:
   - Click the Account Name (an active link) in the Search Results table.
   - Select the account Row and then click **Open**.

The Account: *AccountName* page opens with the target and account information.

From the My Checkouts Page

1. Select **My Checkouts** in the Home accordion.
2. When the My Checkouts page is displayed, locate the account you want to open in the search results table, and then click the Account Name (an active link).

The Account: *AccountName* page opens with the target and account information.

14.5 Checking Accounts Out and In

To check out a privileged account granted to you, follow the instructions provided in Section 9.5, "Checking Out Privileged Accounts."

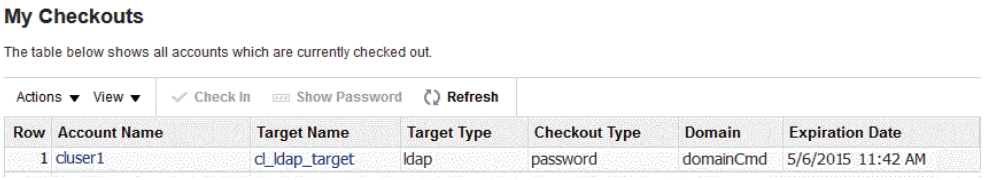
To check an account back in again, follow the instructions provided in Section 9.6, "Checking In Privileged Accounts."

14.6 Viewing Your Checked-Out Accounts

To view a listing of all accounts you currently have checked-out, select **My Checkouts** on the Home accordion.

The My Checkouts page is displayed with all of your checked-out accounts listed in the Search Results table, as shown in Figure 14–1.

Figure 14–1 Example of Checked Out Accounts



14.7 Checking Out Privileged Account Sessions

To check out a privileged account session granted to you, follow the instructions provided in Section 9.5.3, "Checking Out Privileged Account Sessions."

Note: You do not have to perform any special steps to check in a checked out session. If you use the procedure described in Section 9.6, "Checking In Privileged Accounts," then the account is checked back in regardless of the checkout type (password or session).

14.8 Showing a Password

If necessary, you can view the current password in clear text for an account that you have checked out by using the **Show Password** option. For example, if you forget a password, you can use this feature to view the password again.

Any user can view a password for an account they have checked out. However, you cannot access passwords for accounts that are checked in or for accounts that are checked out by other users. Attempts to do so will cause an error.

To view a password, refer to Section 9.8.1, "Showing an Account Password."

Part III

Monitoring Oracle Privileged Account Manager

This part provides information about monitoring Oracle Privileged Account Manager, and it contains the following chapters:

- Working with Reports
- Managing Oracle Privileged Account Manager Auditing and Logging

15

Working with Reports

This chapter describes how to open and work with the Oracle Privileged Account Manager reports by using the Console.

Note:

- You must be an Oracle Privileged Account Manager administrator with the *Security Auditor* Admin Role to open and work with these reports.
- You can also use the Oracle Privileged Account Manager's command line tool or RESTful interface to work with these reports.

  For more information, refer to Appendix A, "Working with the Command Line Tool" and Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface" (respectively).

This chapter includes the following sections:

15.1 Overview

Oracle Privileged Account Manager's real-time reports provide information about the targets and accounts being managed by Oracle Privileged Account Manager. These reports include:

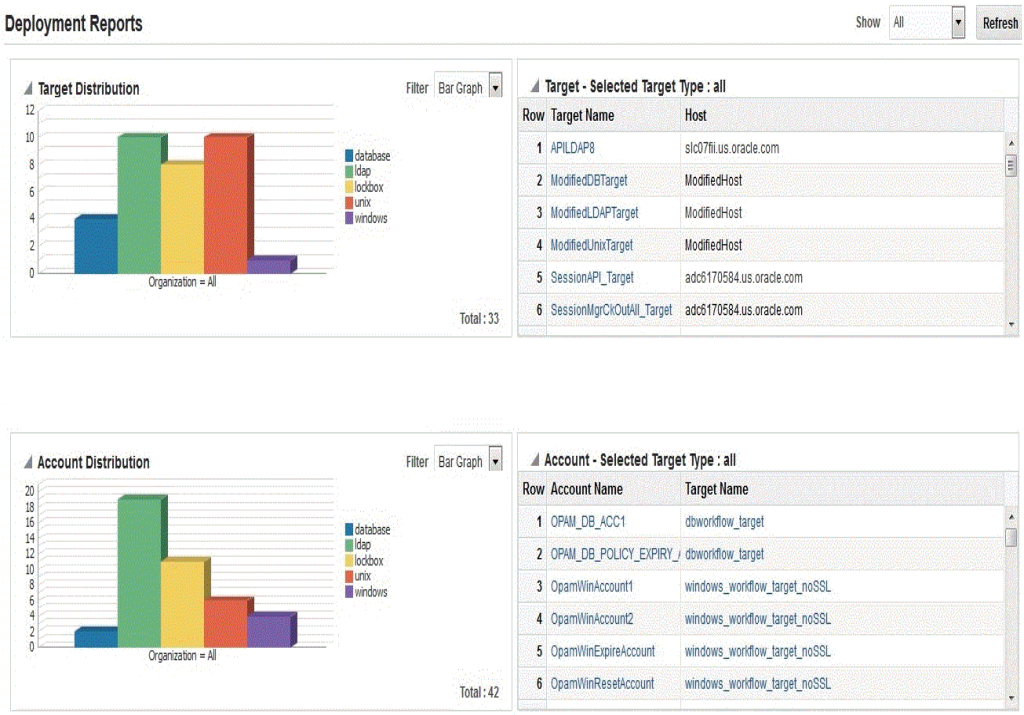
- **Deployment Reports**: Provide information about how targets and privileged accounts are distributed within your deployment.
- **Usage Reports**: Provide information about the privileged accounts that are currently checked out in your deployment.
- **Failure Reports**: Provide information about any on-going target and privileged account failures.
- **Checkout History Reports**: Provide information about the privileged account checkouts that have been performed over a specified period of time.

15.2 Working with Deployment Reports

A Deployment Report provides information about the current state of targets and privileged accounts within your deployment.

To open this report, select the **Deployment Reports** link in the Reports accordion. The Deployment Report page is displayed, similar to the example shown in Figure 15–1.

Figure 15–1 Example Deployment Report



The information on this page is organized into the following portlets:

- The **Target Distribution** and **Account Distribution** portlets provide information about how targets and accounts in a specific organization (or in all organizations) are distributed by target type.
- The **Target** and **Account** portlets show a list of targets and accounts in a specific organization (or in all organizations) by target type.

In addition, you can use the following features on this page to change the data display:

- Use the **Show** menu to manage how much data is displayed in the report:
  - Choose **All** (default) to view the deployment status of targets and accounts for all organizations.
  - Choose the name of an organization to view the deployment status of targets and accounts in that particular organization.
- Use the **Filter** menus to view Target Distribution or Account Distribution data as a **Bar Graph** (default), **Pie Chart**, or **Table**.

  For example, Figure 15–1 shows the Target Distribution data in Bar Graph format. Figure 15–2 and Figure 15–3 show that same data in Pie Chart and Table format (respectively).

Figure 15–2 Example Pie Format

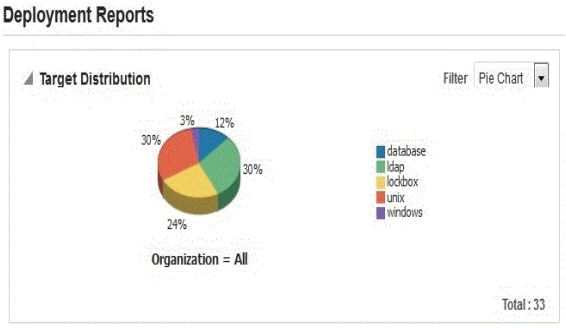
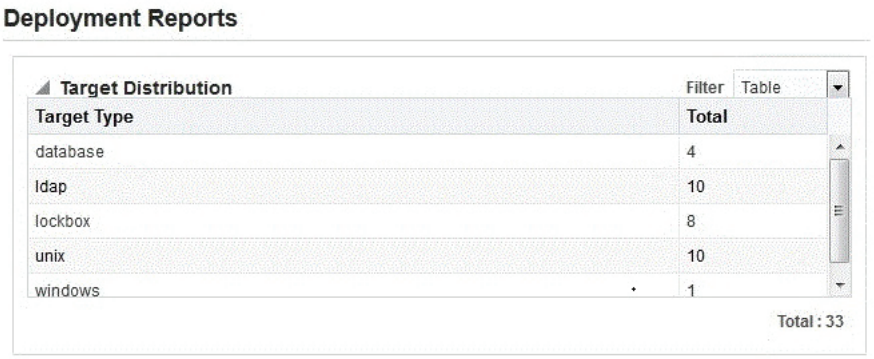


Figure 15–3 Example Table Format



- Hover the cursor over individual pie segments or bars to see a pop-up (such as the one shown in Figure 15–2) with Series, Group, and Value information for that particular Target Type.
  - **Series** identifies which component of the report is reflected by that pie or bar.
  - **Group** indicates the organization (All, ModifiedOrg, Oracle, or ST_Users) on which the report results were based.
  - **Value** indicates how many items were included in that component.
- Click individual pies or bars to limit the Target Deployment or Account Deployment results to a specific Target Type. For example, click the 24% pie in Figure 15–2 and only lockbox targets are displayed in the Target Deployment table.

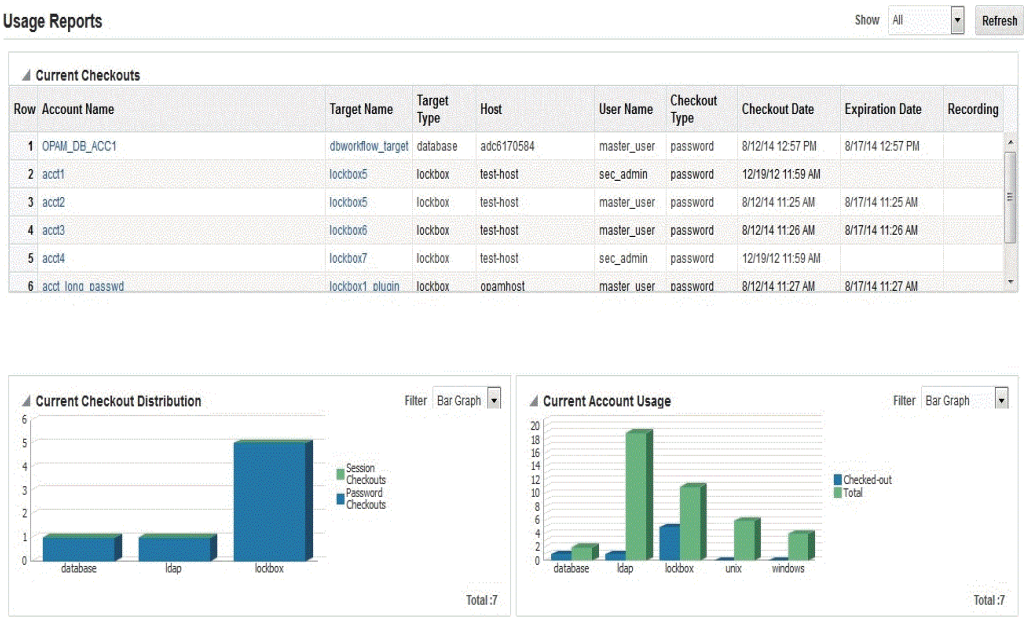
  Click **Refresh** to change the display back to all targets and all accounts.
- Click a Target Name link or an Account Name link in the Deployment tables to open the configuration page for that target or account.

15.3 Working with Usage Reports

A Usage Report provides information about how privileged accounts are currently being used in your deployment.

To open this report, select the **Usage Reports** link in the Reports accordion. The Usage Report page is displayed, similar to the example shown in Figure 15–4.

Figure 15–4 Example Usage Report



The information on this page is organized into the following portlets:

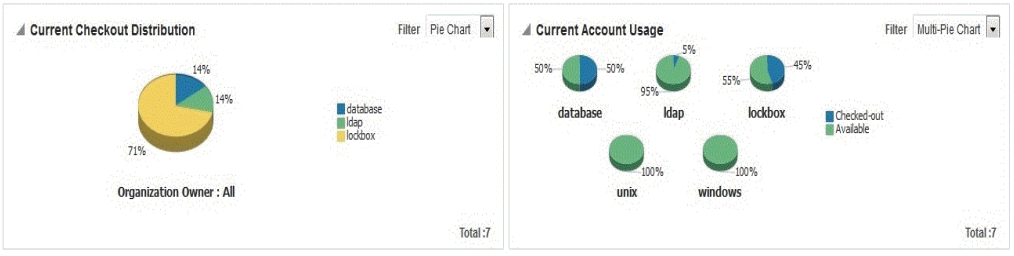
- The **Current Checkouts** portlet summarizes which accounts are currently checked out within your deployment.
- The **Current Checkout Distribution** portlet compares how the current checkouts are distributed, based on checkout type (password or session) and target type.
- The **Current Account Usage** portlet compares how many accounts are currently checked out in relation to the total number of accounts.

In addition, you can use the following features on this page to change the data display:

- Use the **Show** menu to manage how much data is displayed in the report:
  - Choose **All** (default) to view the current account reservations for all organizations.
  - Choose the name of an organization to view the current account reservations for that particular organization.
- Use the **Filter** menus to view the Current Checkout Distribution or Current Account Usage Status data as a **Bar Graph** (default), **Pie Chart**, or **Table**.

  For example, Figure 15–4 shows the Current Account Distribution and the Current Account Usage Status data in the Bar Graph format. Figure 15–2 shows that same data in Pie Chart format.

Figure 15–5 Example Usage Report in Pie Chart Format



- Hover your cursor over individual pies or bars to see a pop-up with Series, Group, and Value information for that particular Target Type.
  - **Series** identifies which component of the report is reflected by that pie or bar.
  - **Group** indicates the organization (All, ModifiedOrg, Oracle, or ST_Users) on which the report results were based.
  - **Value** indicates how many items were included in that component.
- Click a Target Name link or an Account Name link in the Deployment tables to open the configuration page for that target or account.

15.4 Working with Failure Reports

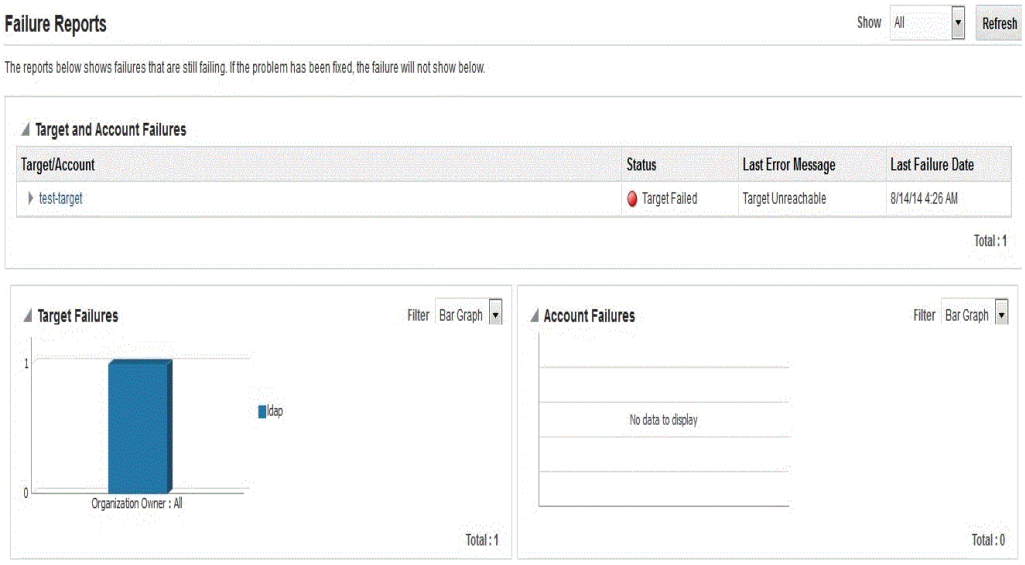
A Failure Report provides information about the current state of target and account failures in your deployment.

Generally, there are two kinds of failure in Oracle Privileged Account Manager:

- **Target failures** occur when Oracle Privileged Account Manager cannot connect to the target, such as when the target server is down. In this situation, Oracle Privileged Account Manager cannot change passwords for any accounts on the failed target.
- **Account failures** occur when Oracle Privileged Account Manager can connect to the account's target, but cannot change the password for that particular account. For example, this failure occurs when accounts get removed from a target.

To open this report, select the **Failure Reports** link in the Reports accordion. The Failure Report page is displayed, similar to the example shown in Figure 15–6.

Figure 15–6 Example Failure Report



The information on this page is organized into the following portlets:

- The **Targets and Accounts Failures** portlet provides a list of targets, the target status, last error message, last failure date, and the total number of target and account failures.

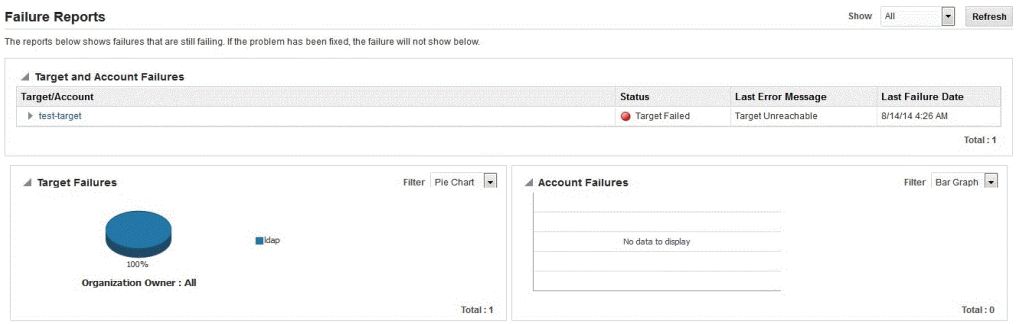
- The **Target Failures** portlet illustrates the distribution of failed targets by target type.
- The **Account Failures** portlet illustrates the distribution of failed accounts by target type.

In addition, you can use the following features on this page to change the data display:

- Use the **Show** menu to manage how much data is displayed in the report:
  - Choose **All** (default) to view all target and account failures.
  - Choose the name of a resource group to view the target and account failures for a particular resource group.
- Use the **Filter** menus to view the Target Failures or Account Failures data as a **Bar Graph** (default), **Pie Chart**, or **Table**.

  For example, Figure 15–6 shows the Target Failures data in Bar Graph Format and the Account Failures data in Table format. Figure 15–7 shows the Target Failures data in Pie Chart format.

Figure 15–7 Example Failure Report in Table and Pie Chart Format



- Hover your cursor over individual pie segments or bars to see a pop-up (such as the one shown in Figure 15–7) with Series, Group, and Value information about that particular failure.
  - **Series** identifies which component of the report is reflected by that pie or bar.
  - **Group** indicates the organization (All, ModifiedOrg, Oracle, or ST_Users) on which the report results were based.
  - **Value** indicates how many items were included in that component.
- Click a **Target/Account** link in the Targets and Accounts Failures table to open the configuration page for that target or account.

15.5 Working with Checkout History Reports

A Checkout History Report provides information about the account checkouts performed over a specified period of time.

Generally, there are three kinds of Checkout History reports in Oracle Privileged Account Manager:

- Plain Text Transcripts:

  The session checkout history recording is available in a plain text transcript format denoted by the ( ) icon.

- Interactive Transcripts:

  The session checkout history recording is available in an interactive transcript format denoted by the ( ) icon. This transcript contains a region where the transcript text is loaded in plain text and another, which contains an outline of all the commands issued to the target system, along with their timestamps. Each command in the outline will link to a relevant region of the transcript, when clicked.

  When you click a command on the "Command List", the session transcript will move to the position in the transcript where that command is executed.

- Video Recordings:

  The session checkout history recording is available in a video recording format denoted by the ( ) icon. Video recordings are available for Windows targets only.

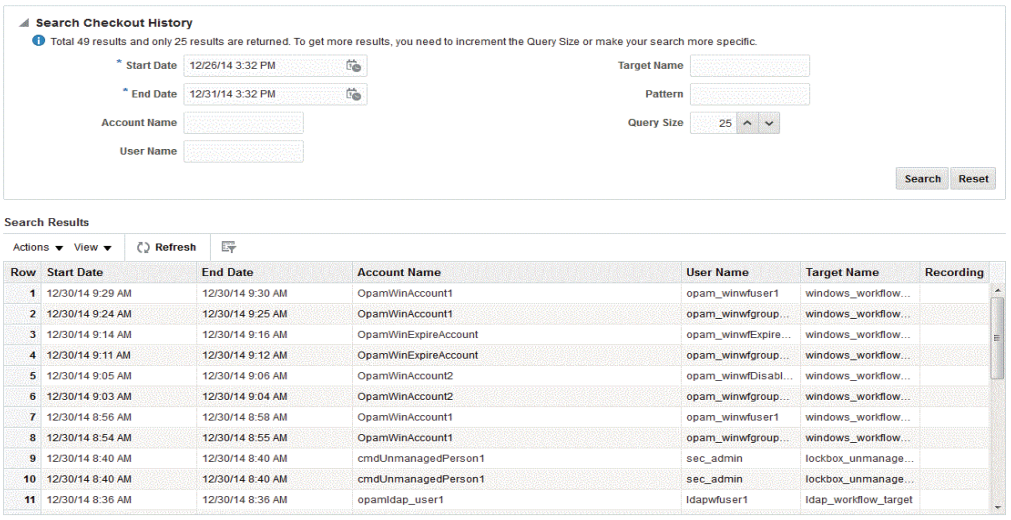
  In the video recording re-play page, when you click an event in the "Event Index," the video move to part of the video where the event begins and pause there.

To open a Checkout History report,

1. Select the **Checkout History Report** link in the Reports accordion.

   The Checkout History Report page is displayed, similar to the example shown in Figure 15–8.

Figure 15–8 Example Checkout History Report



2. Use the Search Checkout History portlet to configure your search parameters:

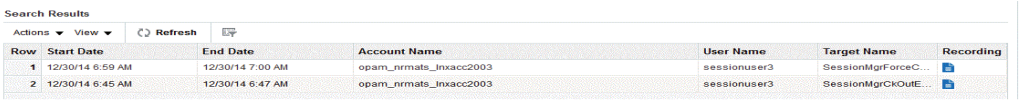
   - You must specify a **Start Date** and an **End Date** range in which to search for checkouts. Type a date and time into the blank fields or use the **Calendar** icons.

   - Enter information into one or more of the **Account Name**, **User Name**, **Target Name**, or **Pattern** fields to further narrow the search results. (*Optional*)

Note:

- Use the **Pattern** field to search for a string in the session recording of a checkout event. For example, entering **`rmdir`** narrowed the search results in Figure 15–8 to the two entries as shown in Figure 15–9.
- Oracle Text indexes are used for pattern search. To maintain these indexes refer to Section 17.5.5, "Managing Oracle Text Index for Session Recordings."
- If the recent session recordings are not listed in the pattern search results, refer to Section 20.3.22, "Checkout History Search Results for Pattern Search Do Not Include Recent Session Recordings" to troubleshoot this issue.

Figure 15–9 Example Search Results



- Enter a value into the **Query** field to limit the number of returned results.

3. Click **Search** and the results are displayed in a table that is organized into the following columns:
   - **Start Date** and **End Date**: Date that the account was checked out and checked back in, respectively.
   - **Account Name**: Name of the checked out account.
   - **User Name**: Name of the user who checked out the account.
   - **Target Name**: Name of the target associated with the account.
   - **Recording**: The recording icon or icons in this column indicate that the recording of the user's actions during the checkout is available for viewing. You must click the icon to view the specific type of recording.

     For a unix target, the recording can be in the plain text transcript format denoted by the ( ) icon, or the interactive transcript format denoted by the ( ) icon. For a windows target, the recording is available in the video format denoted by the ( ) icon.

     Note: Only an administrator can view session recordings. Refer to Section 9.7, "Viewing a Session Recording" for more information.

16

Managing Oracle Privileged Account Manager Auditing and Logging

This chapter describes how to configure and use Oracle Privileged Account Manager's auditing and logging functionality.

This chapter includes the following sections:

- Section 16.1, "Understanding Oracle Privileged Account Manager Auditing"
- Section 16.2, "Understanding Oracle Privileged Account Manager Logging"

> **Note:** If you are using Oracle Privileged Account Manager on IBM WebSphere, refer to "Differences in Managing Oracle Privileged Account Manager Auditing and Logging" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for information about this topic.

16.1 Understanding Oracle Privileged Account Manager Auditing

Oracle Privileged Account Manager audits all security events that occur under its purview, which gives you better visibility into how privileged accounts are used within your organization and enables you to effectively manage sensitive information.

Specifically, the Oracle Privileged Account Manager audit logger logs any events that modify entity states; such as when you add, modify, or remove new accounts, targets, or policies.

The following table describes all of the event categories and event types for which an audit can be generated:

Table 16–1 Audited Oracle Privileged Account Manager Events

Event Category	Event Types	Description
Account Management		Events related to managing *principal* accounts **Note**: A principal can be an end-user or a pseudo-user (a service within the system).
	Add Account	Adding users, groups, or any other principal accounts
	Change Password	Changes to user passwords
	Disable Account	Disabling users, groups, or any other principal accounts

Table 16–1 (Cont.) Audited Oracle Privileged Account Manager Events

Event Category	Event Types	Description
	Enable Account	Enabling users, groups, or any other principal accounts
	Modify Account	Modifying account attributes
	Query Account	Queries to a user's account
	Remove Account	Removing users, groups, or any other principal accounts
Policy Management		Events related to managing policies
	Create Policy	Creating policies
	Delete Policy	Deleting policies
	Modify Policy	Modifying policies
	Query Policy	Querying policies
Target Management		Events related to managing targets
	Add Target	Adding targets
	Modify Target	Modifying targets
	Query Target	Querying targets
	Remove Target	Removing targets

Logging these audit events creates a processing history that allows reporting tools to gather statistics, as described in section Section 16.1.2, "Understanding Oracle Privileged Account Manager Audit Reports"

This section discusses the following topics:

- Section 16.1.1, "Configuring Auditing in Oracle Privileged Account Manager"
- Section 16.1.2, "Understanding Oracle Privileged Account Manager Audit Reports"
- Section 16.1.3, "Auditing Application Consumption of Credentials from CSF"

16.1.1 Configuring Auditing in Oracle Privileged Account Manager

You can configure Oracle Privileged Account Manager to save audit events into a database or a file. When a database is not available, Oracle Privileged Account Manager saves its audit logs into this file,

```
DOMAIN_HOME/servers/<opamserver>/logs/auditlogs/OPAM
```

You can also configure Oracle Privileged Account Manager to deploy audit reports in BI Publisher (version 11.1.1.5.0 or higher), and use BI Publisher to view audit events in the database. Reports in BI Publisher are only possible if the audit events are being pushed into a database and not a file.

The following topics provide instructions for configuring auditing in Oracle Privileged Account Manager:

- Section 16.1.1.1, "Configuring File-Based Auditing in Oracle Privileged Account Manager"
- Section 16.1.1.2, "Configuring Database-Based Auditing in Oracle Privileged Account Manager"

- Section 16.1.1.3, "Deploying Oracle Privileged Account Manager Audit Reports in BI Publisher"
- Section 16.1.1.4, "Setting the Audit Logging Levels"

> **Note:** To configure auditing for Oracle Privileged Account Manager on an IBM WebSphere server, refer to "Configuring Auditing for Oracle Privileged Account Manager" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* before starting the procedures described in this section.

16.1.1.1 Configuring File-Based Auditing in Oracle Privileged Account Manager

This section describes how to configure file-based auditing in Oracle Privileged Account Manager.

Before You Begin

Before starting the following configuration steps, review these publications:

- "Using WLST Online or Offline" in the *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*
- "OPSS Scripts for Auditing" in the *Oracle Fusion Middleware Application Security Guide* for detailed information about the `getAuditPolicy`, `setAuditPolicy`, `getAuditRepository`, and `setAuditRepository` WLST audit commands used in the configuration steps.

To configure Oracle Privileged Account Manager for file-based auditing:

1. Start the WebLogic Scripting Tool (WLST) and connect to the Oracle WebLogic Server:
   a. Open a command window and navigate to the following directory, which contains the WLST:

      MW_HOME`/oracle_common/common/bin`

   b. Start WLST by typing one of the following commands:

      On UNIX, type: `sh wlst.sh`

      On Windows, type: `wlst.cmd`

      You know that WLST has started when the command prompt changes to `wls:>/offline`.

   c. Connect to the Oracle WebLogic Server by typing the following command:

      `connect('`*WLS_Admin_Name*`','`*WLS_Admin_Password*`','`*WLS_Machine_Name*`:`*Port*`')`

      For example,

      `connect('weblogic','password1','`*localhost*`:7001')`

      WLST validates the administrator's username and password, the machine name, and the port that are associated with the WebLogic Admin Server. If all of these values are correct, WLST connects to the WebLogic Admin Server and the command prompt changes to

      `wls:>/base_domain/serverConfig`

Note: Refer to "Securing Access from WLST Online" in the *Oracle Fusion Middleware Oracle WebLogic Scripting Tool* for additional information.

2. To set the audit logging level for Oracle Privileged Account Manager:
   a. If the `filterPreset` parameter is set to `NONE`, use the `setAuditPolicy` command to change the value to `All`, `Medium`, or `Low`, based on how much logging you want Oracle Privileged Account Manager to provide:

      ```
      setAuditPolicy(filterPreset='All')
      ```

      A confirmation message is displayed to indicate the audit logging level was successfully updated.

      Note: For a description of the different logging levels, refer to the table Step 2 of Section 16.1.1.4, "Setting the Audit Logging Levels."

   b. Verify the current logging level for Oracle Privileged Account Manager, by typing `getAuditPolicy( )` at the prompt, and then checking the `filterPreset` parameter value.
3. To change the Repository Type to database (`DB`):
   a. Type the `setAuditRepository` command as follows:

      ```
      setAuditRepository(switchToDB='true')
      ```

      A confirmation message is displayed to let you know that the audit repository was successfully updated.

   b. You can use the WLST `getAuditRepository` command to verify that the audit repository is set to database-based auditing:

      ```
      getAuditRepository( )
      ```

      The `setAuditRepository` parameter value (as indicated by the Repository Type field) should be **`FILE`**.
4. Restart both the Administration Server and the Oracle Privileged Account Manager Managed Server.

   Note: For detailed information about starting a Managed Server, refer to "Starting or Stopping the Oracle Stack" in the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*.

   You must restart both servers for your changes to take effect. After the server restarts, audit logs will start appearing in this location:

   DOMAIN_HOME`/servers/<`*opamserver*`>/logs/auditlogs/OPAM`

16.1.1.2 Configuring Database-Based Auditing in Oracle Privileged Account Manager

This section describes how to configure Oracle Privileged Account Manager to save audit events into the Oracle database that is associated with Oracle Privileged Account

Manager.

Prerequisites

Before starting the following configuration steps,

- Review these publications:
  - "Using WLST Online or Offline" in the *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*
  - "WLST Commands for Auditing" in the *Oracle Fusion Middleware Application Security Guide* for detailed information about the `getAuditPolicy`, `setAuditPolicy`, `getAuditRepository`, and `setAuditRepository` WLST audit commands used in the configuration steps.
- Install the following
  - A database
  - The Repository Creation Utility application, which is used to create a schema and load a repository into the database.

> **Note:** For information about installing and working with the Repository Creation Utility, refer to *Oracle Fusion Middleware Repository Creation Utility User's Guide* available at `http://www.oracle.com/technology/documentation/index.html`

To configure database-based auditing:

1. Start the WebLogic Scripting Tool (WLST) and connect to the Oracle WebLogic Server:
   a. Open a command window and navigate to the following directory, which contains the WLST:

      MW_HOME`/oracle_common/common/bin`

   b. Start WLST by typing one of the following commands:

      On UNIX, type: `sh wlst.sh`

      On Windows, type: `wlst.cmd`

      You know that WLST has started when the command prompt changes to `wls:>/offline`.

   c. Connect to the Oracle WebLogic Server by typing the following command:

      `connect('`*WLS_Admin_Name*`','`*WLS_Admin_Password*`','`*WLS_Machine_Name*`:`*Port*`')`

      For example,

      `connect('weblogic','password1','`*localhost*`:7001')`

      WLST validates the administrator's username and password, the machine name, and the port that are associated with the WebLogic Admin Server. If all of these values are correct, WLST connects to the WebLogic Admin Server and the command prompt changes to

      `wls:>/base_domain/serverConfig`

Note: Refer to "Securing Access from WLST Online" in the *Oracle Fusion Middleware Oracle WebLogic Scripting Tool* for additional information.

2. To set the audit logging level for Oracle Privileged Account Manager:

   a. If the `filterPreset` parameter is set to `NONE`, use the `setAuditPolicy` command to change the value to `All`, `Medium`, or `Low`, based on how much logging you want Oracle Privileged Account Manager to provide:

   ```
   setAuditPolicy(filterPreset='All')
   ```

   A confirmation message is displayed to indicate the audit logging level was successfully updated.

   Note: For a description of the different logging levels, refer to the table in Step 2 of Section 16.1.1.4, "Setting the Audit Logging Levels".

   b. Verify the current logging level for Oracle Privileged Account Manager, by typing `getAuditPolicy( )` at the prompt, and then checking the `filterPreset` parameter value.

3. To change the Repository Type to database (`DB`):

   a. Type the `setAuditRepository` command as follows:

   ```
   setAuditRepository(switchToDB='true')
   ```

   A confirmation message is displayed to let you know that the audit repository was successfully updated.

   b. You can use the WLST `getAuditRepository` command to verify that the audit repository is set to database-based auditing:

   ```
   getAuditRepository( )
   ```

   The `setAuditRepository` parameter value (as indicated by the Repository Type field) should be **`DB`**.

4. Use the Repository Creation Utility to create and load the audit schema into the database, and then use the WebLogic Server Administrative Console to create a new JDBC data source.

   A *data source* contains credentials that BI Publisher needs to connect to the Oracle database associated with Oracle Privileged Account Manager. BI Publisher uses this connection to retrieve data from the Oracle Privileged Account Manager database. BI Publisher then uses this data to generate reports for targets, privileged accounts, grants, and policies.

   Note: Instructions for creating the audit schema and for creating a JDBC data source are provided in the "Configuring and Managing Auditing" section of the *Oracle Fusion Middleware Application Security Guide*.

5. Restart both the Administration Server and the Oracle Privileged Account Manager Managed Server.

   You must restart both servers for your changes to take effect. After restarting both servers, audit logs will start appearing in the installed database.

16.1.1.3 Deploying Oracle Privileged Account Manager Audit Reports in BI Publisher

This section describes how to deploy Oracle Privileged Account Manager audit reports in Oracle Business Intelligence Publisher (BI Publisher), a component used to manage and deliver reports.

Use the following steps:

1. Install and configure BI Publisher version 11.1.1.5.0 or higher if it is not already installed.
2. After installing BI Publisher, locate the following directory in the WebLogic domain:

   Note: You can deploy BI Publisher on the same host or in a different domain.

   ```
   BI_DOMAIN_HOME/config/bupublisher/repository/Reports
   ```

3. Locate the `opam_product_BIP11gReports_11_1_2_1_0.zip` file in the following directory:

   ```
   ORACLE_HOME/opam/reports
   ```

   Unzip this file into the `Reports` folder noted in step 2 and verify that the following directory was created:

   ```
   ORACLE_HOME/opam/reports/Oracle Privileged Account Manager
   ```

4. To set up the catalog and configure data sources, open a browser window and enter the URL for BI Publisher.

   The format for this URL is

   http://*hostname: port/xmlpserver/*

   For example

   http:/*localhost:2001/xmlpserver/*

5. When the BI Publisher login page is displayed, log in as a user with WebLogic privileges and click **Sign In.**
6. Set up the catalog as follows:
   a. Select **Administration** > **System Maintenance** > **Server Configuration.**
   b. When the System Maintenance page is displayed, go to the **Path** field in the **Configuration Folder** section and enter the path to your Configuration folder. For example,

      ```
      BI_DOMAIN_HOME/config/bupublisher/repository
      ```

      The files that contain your server configuration settings (such as the JDBC data source you created in step 4 of Section 16.1.2) are stored in a Configuration folder. The path to this folder is stored in the xmlp-server-config.xml configuration file. The xmlp-server-config.xml file is located in

      ```
      BI_DOMAIN_HOME/config/bupublisher/repository/Admin/Configuration
      ```

c. Locate the **Catalog** section on the System Maintenance page and specify the following information:

Parameter Name	Parameter Value
Catalog Type	Select **BI Publisher - File System** from the menu.
Path	Enter the path to the BI Publisher Catalog folder. For example, *BI_DOMAIN_HOME/config/bipublisher/repository* **Caution:** The path to the BI Publisher Catalog includes the `reports` subdirectory where you unpacked the Oracle Privileged Account Manager reports. **Do not** include the `reports` subdirectory in the Path field or you will corrupt BI Publisher.

Note:

Because the file system contains the reports repository, the platform where you are running BI Publisher determines the case-sensitivity of folder and report names. Repository object names are not case-sensitive in a Windows-based environment, but they are case-sensitive in a UNIX-based environment.

d. Click **Apply.**

A confirmation message is displayed.

e. Log in as an administrator.

f. Click **Catalog** to open the `Shared Folder/ Oracle Privileged Account Manager` folder.

Note:

If this folder does not display, restart the application from the WebLogic console.

7. One JDBC (Oracle Privileged Account Manager JDBC) connection is required for Oracle Privileged Account Manager reports. Use the following steps to define an Oracle Privileged Account Manager JDBC connection and define the data sources:

a. Click the Administration link found on the right side of the BI Publisher page.

The BI Publisher Administration page is displayed. **Note:** Notice the Data Sources section on this page.

b. Click the **JDBC Connection** link found in the Data Sources section.

c. When the Data Sources page is displayed, click Add Data Source in the JDBC section to create a JDBC connection to your database.

d. On the Add Data Source page, enter the following information:

Parameter Name	Parameter Value
Data Source Name	`OPAM JDBC`
Driver Type	Select a driver type to suit your database (for example, Oracle 10g or Oracle 11g).
Database Driver Class	`oracle.jdbc.driver.OracleDriver` You must define a driver class to suit your database.
Connection String	Provide the database connection details. For example, *hostname:port:sid*.
User name	Provide the Oracle Privileged Account Manager Audit DB user name.
Password	Provide the Oracle Privileged Account Manager Audit DB user password.

If the connection to the database is established, a confirmation message is displayed indicating the success.

e. Click **Apply.**

You should see this newly defined connection (Oracle Privileged Account Manager JDBC) in the list of JDBC Data Sources.

f. Navigate to Oracle Privileged Account Manager Audit Reports.

The Catalog page is displayed as a tree structure on the left side of the page with details on the right.

g. Expand `Shared Folders` and select the `Oracle Privileged Account Manager` folder to view all of the objects in that folder.

You can now navigate in BI Publisher and use the Oracle Privileged Account Manager 11g BI reports.

16.1.1.4 Setting the Audit Logging Levels

To change the amount of audit logging provided by Oracle Privileged Account Manager, use the following steps:

1. Launch an application server shell (WLST) and establish a connection to the Oracle WebLogic Server as described in step 4 of Section 16.1.1.2, "Configuring Database-Based Auditing in Oracle Privileged Account Manager."

   Note: Refer to "Securing Access from WLST Online" in the *Oracle Fusion Middleware Oracle WebLogic Scripting Tool* for more information.

2. Use the `getAuditPolicy` command to get the current audit policy.

   If the FilterPreset field is set to `NONE`, use the `setAuditPolicy` command to change the value. Choose one of the options noted the following table, depending on the type of events to be audited:

   Note: Refer to "getAuditPolicy" and "setAuditPolicy" in the *Oracle Fusion Middleware Application Security Guide* for detailed information about these WLST audit commands.

Option	Logged Events
All	Logs all event types.
Medium	Logs the following event types: ■ **In the AccountManagement category**: ChangePassword, CheckinAccount, CreateAccount, DeleteAccount, DisableAccount, EnableAccount, ModifyAccount, and QueryAccount ■ **In the PolicyManagement category**: All ■ **In the TargetManagement category**: All
Low	Logs the following event types: ■ **In the AccountManagement category**: ChangePassword, CheckinAccount, CreateAccount, DeleteAccount, DisableAccount, EnableAccount, and ModifyAccount ■ In the PolicyManagement category: CreatePolicy, DeletePolicy, and ModifyPolicy ■ In the TargetManagement category: CreateTarget, DeleteTarget, and ModifyTarget
None	No logging is performed.

3. Restart the Oracle Privileged Account Manager server.

> **Note:** For detailed information about starting a Managed Server, refer to "Starting or Stopping the Oracle Stack" in the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*.

After the server restarts, audit logs will start appearing in this location:

DOMAIN_HOME/servers/<*opamserver*>/logs/auditlogs/OPAM

16.1.2 Understanding Oracle Privileged Account Manager Audit Reports

Oracle Privileged Account Manager supplies a set of default audit reports that are integrated with BI Publisher 11g and the Oracle Fusion Middleware Audit Framework. Oracle Privileged Account Manager generates these reports based on the audit events logged in the audit store.

The default audit report types include:

- **Accounts Checkin Checkout Report**: Provides account checkout and check-in history.
- **All Events Report**: Includes all audit events that are logged in the audit store.
- **Error Events Report**: Provides information about any errors that occur in Oracle Privileged Account Manager, such as authentication and authorization failures.
- **General Report**: Provides information about events related to checking in, checking out, or modifying privileged accounts and events related to queries about privileged accounts and targets.
- **Target Management Report**: Provides information about events related to adding, modifying, querying, or removing targets.

Oracle Privileged Account Manager audit reports can show who checked out an account and on which system it was checked out, justifications, requests for a system

that is already checked out, and requests for a system to which a user does not have privileges.

For example, the following figure shows a typical Oracle Privileged Account Manager audit report as viewed in BI Publisher.

> **Note:** You can view Oracle Privileged Account Manager audit reports in BI Publisher.

Figure 16–1 Example Oracle Privileged Account Manager Audit Report

Event	Status	User ID	Target	Resource ID	Message	Time
CheckoutAccount	1	Chris Paul	demoDB:DEV_IAU_APPEND	7f3fa17504f64c7c806fbe8336e2dc27	Checkout Account: demoDB:DEV_IAU_APPEND:7f3fa17504f64c7c806fbe8336e2dc27[NO_COMMENTS_PROVIDED]	1/16/13 12:14 PM Midway
CheckoutAccount	1	Les Paul	demoDB:DEV_IAU_APPEND	7f3fa17504f64c7c806fbe8336e2dc27	Checkout Account: demoDB:DEV_IAU_APPEND:7f3fa17504f64c7c806fbe8336e2dc27[NO_COMMENTS_PROVIDED]	1/16/13 12:15 PM Midway
CheckoutAccount	1	Tak Matsumoto	demoDB:DEV_OPSS	7c3c3731d930442492ef5759a64e164c	Checkout Account: demoDB:DEV_OPSS:7c3c3731d930442492ef5759a64e164c[NO_COMMENTS_PROVIDED]	1/16/13 12:16 PM Midway
CheckinAccount	1	Tak Matsumoto	demoDB:DEV_OPSS	7c3c3731d930442492ef5759a64e164c	Checkin Account: demoDB:DEV_OPSS:7c3c3731d930442492ef5759a64e164c	1/16/13 12:16 PM Midway
CheckinAccount	1	Chris Paul	demoDB:DEV_IAU_APPEND	7f3fa17504f64c7c806fbe8336e2dc27	Checkin Account: demoDB:DEV_IAU_APPEND:7f3fa17504f64c7c806fbe8336e2dc27	1/16/13 12:17 PM Midway
CheckoutAccount	1	Chris Paul	demoDB:DEV_IAU_APPEND	7f3fa17504f64c7c806fbe8336e2dc27	Checkout Account: demoDB:DEV_IAU_APPEND:7f3fa17504f64c7c806fbe8336e2dc27[NO_COMMENTS_PROVIDED]	1/16/13 12:17 PM Midway
CheckoutAccount	1	Les Paul	lockbox1:account1	3ed20fc11ac540a1b7dc4937ef0a5b8c	Checkout Account: lockbox1:account1:3ed20fc11ac540a1b7dc4937ef0a5b8c[NO_COMMENTS_PROVIDED]	1/16/13 12:17 PM Midway
CheckinAccount	1	Les Paul	demoDB:DEV_IAU_APPEND	7f3fa17504f64c7c806fbe8336e2dc27	Checkin Account: demoDB:DEV_IAU_APPEND:7f3fa17504f64c7c806fbe8336e2dc27	1/16/13 12:17 PM Midway

Notice that this report provides the following information:

- **Event**: Type of event that occurred
- **Status**: Event results, where 1 is success and 0 is a failure
- **User ID**: User that initiated the event
- **Target**: Target on which the event occurred
- **Resource ID**: Resource identifier
- **Message**: Message returned from server
- **Time**: Date and time the event occurred

> **Note:** Not all fields in the IAU_COMMON table are used. Only the fields that are relevant to the audited data are populated.

16.1.3 Auditing Application Consumption of Credentials from CSF

Oracle Privileged Account Manager can synchronize passwords to CSF, as described in Section 19.3, "Integrating with the Credential Store Framework." However, Oracle Privileged Account Manager cannot audit any CSF content because Oracle Privileged Account Manager and CSF are two separate entities in the WebLogic domain. If you want to audit CSF access, then you must enable auditing in CSF itself.

Note: For information about enabling auditing in CSF, refer to the following sections in the *Oracle Fusion Middleware Application Security Guide*:

- For a list of the audit events that are supported by CSF, refer to "Oracle Platform Security Services Events and their Attributes."
- For information about the WLST commands used to enable auditing in CSF, refer to "WLST Commands for Auditing" or enter the following command from the command line:

  ```
  help('<Audit WLST command>')
  ```

- For information about using Enterprise Manager to manage this type of auditing, refer to "Managing Audit Policies."

For information about using WSAdmin commands to enable auditing in CSF, refer to "Executing Common Audit Framework wsadmin Commands" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management*.

16.2 Understanding Oracle Privileged Account Manager Logging

This section discuss the following topics:

- Section 16.2.1, "Configuring Basic Logging"
- Section 16.2.2, "Example Logging Data"

Oracle Privileged Account Manager is fully integrated with Oracle Fusion Middleware Logging and the Oracle Diagnostic Logging (ODL) framework.

The Oracle Privileged Account Manager generic logger (oracle.idm.opam) takes care of all logs not recorded by the audit logger, which includes debugging statements and exception messages. Processing tools can use these logs to diagnose problems that occur within the Oracle Privileged Account Manager server.

Table 16–2 describes the different Oracle Privileged Account Manager-related log files:

Table 16–2 Oracle Privileged Account Manager-Related Log Files

File Name	Description
`AdminServer.log`	Generic log file where the WebLogic Admin Server writes messages from its subsystems and applications.
`AdminServer-diagnostic.log`	Diagnostic log file used to store messages generated by the WebLogic Admin Server.
`base_domain.log`	Generic log file where the WebLogic Admin Server writes messages about the overall status of the domain.
`access.log`	Generic log file used to store information about requests to access privileged accounts and targets.
`opam_server1.log`	Generic log file where the Oracle Privileged Account Manager Server writes messages from its subsystems and applications.
`opam_server1-diagnostic.log`	Diagnostic log file used to store messages generated by the Oracle Privileged Account Manager Server.

Oracle Privileged Account Manager log files are stored in the following locations:

- Server log files are stored in

DOMAIN_HOME/servers/*OPAM managed server*/logs

Server application logging is spooled to

OPAM managed server-diagnostic.log

- Console log files are stored in

  DOMAIN_HOME/servers/AdminServer/logs

> **Note:** For more information about Oracle Fusion Middleware Logging and the Oracle Diagnostic Logging (ODL) framework, refer to "Managing Log Files and Diagnostic Data" in the *Oracle Fusion Middleware Administrator's Guide*.

16.2.1 Configuring Basic Logging

You can configure Oracle Privileged Account Manager logging by using the standard WLST commands as described in "Logging Custom WLST Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

Following are some task-based invocations based on the preceding reference:

> **Note:** The same commands apply if you are configuring logging on an IBM WebSphere server, however there are some differences to consider.
>
> Before using these commands, refer to "Configuring Basic Logging for Oracle Privileged Account Manager" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management*.

- To list all of the available Oracle Privileged Account Manager loggers and their current configured levels, run the **`listLoggers`** command:

  ```
  listLoggers(target="<opamserver>",pattern="oracle.idm.opam.*")
  ```

  For example,

  ```
  listLoggers(target="opam_server1",pattern="oracle.idm.opam.*")
  ```

- To check Oracle Privileged Account Manager's current log level, run the **`getLogLevel`** command:

  ```
  getLogLevel(logger="oracle.idm.opam",target="<opamserver>")
  ```

  For example,

  ```
  getLogLevel(logger="oracle.idm.opam",target="opam_server1")
  ```

- To set the log level for a particular logger, run the **`setLogLevel`** command:

  ```
  setLogLevel(target="<opamserver>",logger="oracle.idm.opam",level="TRACE:32",
  persist=1)
  ```

  For example,

  ```
  setLogLevel(target="opam_server1",logger="oracle.idm.opam",level="TRACE:32",
  persist=1)
  ```

16.2.2 Example Logging Data

This figure shows some example logging data as viewed from the WebLogic console.

Figure 16–2 Example Logging Report

	Date	Subsystem	Severity	Message ID	Message
○	Oct 13, 2011 10:48:25 AM PDT	OPAM	Info	BEA-000000	UIResource/getAccount
○	Oct 13, 2011 10:48:27 AM PDT	OPAM	Info	BEA-000000	PrivilegedAccountResource/updateAccount
○	Oct 13, 2011 10:48:28 AM PDT	OPAM	Info	BEA-000000	UIResource/getAccount
○	Oct 13, 2011 10:48:44 AM PDT	OPAM	Info	BEA-000000	PrivilegedAccountResource/checkout
○	Oct 13, 2011 10:48:48 AM PDT	OPAM	Info	BEA-000000	ContextManager added session 7920930130191964with result = true
○	Oct 13, 2011 10:50:26 AM PDT	OPAM	Info	BEA-000000	UIResource/getAllCheckedOutAccounts
○	Oct 13, 2011 10:50:35 AM PDT	OPAM	Info	BEA-000000	PrivilegedAccountResource/checkIn
○	Oct 13, 2011 10:50:39 AM PDT	OPAM	Info	BEA-000000	ContextManager removed session 7920930130191964with result = true

Notice that this report provides the following information:

- Date and timestamp when the event occurred
- Subsystem on which the event occurred
- Message severity
- Message ID
- Message describing the operation that was performed

Part IV

Advanced Administration

This part provides information about performing advanced administration tasks for Oracle Privileged Account Manager, and it contains the following chapters:

- Performing Advanced Configuration Tasks for Oracle Privileged Account Manager
- Developing Plug-Ins for Oracle Privileged Account Manager
- Integrating Oracle Privileged Account Manager with Other Oracle Identity Management Components
- Troubleshooting Oracle Privileged Account Manager

17

Performing Advanced Configuration Tasks for Oracle Privileged Account Manager

This chapter provides information about performing some advanced configuration for Oracle Privileged Account Manager.

This chapter includes the following sections:

Note: If you are using Oracle Privileged Account Manager on IBM WebSphere, refer to "Differences in Performing Advanced Configuration Tasks for Oracle Privileged Account Manager on IBM WebSphere" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for information about this topic.

17.1 Configuring Oracle Privileged Account Manager to Communicate With Target Systems Over SSL

Oracle Privileged Account Manager can connect to target systems through Secure Socket Layer (SSL) or non-SSL options. The SSL option is more secure, but requires some additional configuration.

To communicate securely over SSL with a target system, the WebLogic instance running Oracle Privileged Account Manager must trust the SSL certificate used by the target system because Oracle Privileged Account Manager inherits its SSL configuration from the WebLogic container in which it runs. To have the WebLogic instance running Oracle Privileged Account Manager (and therefore Oracle Privileged Account Manager) trust the target system's SSL certificate, you must import the certificate into the truststore used by that WebLogic instance.

Note: The steps for configuring SSL communication are different if you are using an IBM WebSphere instance.

Refer to "Differences When Configuring Oracle Privileged Account Manager to Communicate with Target Systems Over SSL" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for instructions.

Use the following steps to enable SSL communication between the target system and Oracle Privileged Account Manager:

1. Export the SSL certificate from the target system host computer.

   Note: The steps for exporting an SSL certificate are different for each target system type. Refer to the product documentation provided for your target system for detailed instructions.

2. Copy the certificate to the machine where you have the WebLogic instance running Oracle Privileged Account Manager.

   If you have the Oracle Privileged Account Manager Console and the Oracle Privileged Account Manager server running on different machines, you must copy the SSL certificate to the Oracle Privileged Account Manager server machine.

3. Run the following command to import the certificate into the JVM truststore of the WebLogic Server on which Oracle Privileged Account Manager is running:

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

   Where

   - *JAVA_HOME* is the location used by your WebLogic server. For example.
     - *MW_HOME*/jrockit..
     - *MW_HOME*/jdk..
     - The location where you installed the Java software
   - *FILE_LOCATION* is the full path and name of the certificate file.
   - *TRUSTSTORE_LOCATION* is one of the following truststore paths:

Table 17–1 Truststore Locations

If you are using:	Import the Certificate into the Keystore in This Directory:
Oracle jrockit_R27.3.1-jdk	*JROCKIT_HOME*/jre/lib/security
The default Oracle WebLogic Server JDK	*WEBLOGIC_HOME*/java/jre/lib/security/cacerts
A JDK other than Oracle jrockit_R27.3.1-jdk or Oracle WebLogic Server JDK	*JAVA_HOME*/jre/lib/security/cacerts

   - *TRUSTSTORE_PASSWORD* is the password for the truststore.
   - *ALIAS* is an alias for the certificate.

Note: The default password for the `cacerts` keystore is *changeit*.

4. Restart all WebLogic servers.

Note: For more information about WebLogic security concepts and how to create custom keystores, refer to "Configuring Identity and Trust" in the *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

17.2 Securing Data On Disk

Oracle Privileged Account Manager can operate with or without Oracle Database Transparent Data Encryption (TDE) mode. This feature enables you to protect sensitive data stored in operating system files by encrypting it. Then, to prevent unauthorized decryption, it stores encryption keys in an external security module. Transparent Data Encryption provides the assurance that the data is encrypted, yet handling encrypted data becomes transparent to applications.

Note: Oracle *strongly recommends* that you enable TDE mode for enhanced security.

For more information about Transparent Data Encryption, refer to the "Securing Stored Data Using Transparent Data Encryption" topic in *Oracle Database Advanced Security Administrator's Guide*.

You can enable or disable TDE mode at any point after installing and configuring Oracle Privileged Account Manager.

This section describes how to change the TDE mode for Oracle Privileged Account Manager. The topics include:

- Section 17.2.1, "Enabling TDE Mode"
- Section 17.2.2, "Disabling TDE Mode"

Note: The instructions for enabling or disabling TDE mode are essentially the same whether you are using a WebLogic server or an IBM WebSphere server.

Refer to "Differences When Securing Data On Disk" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for information about the minor differences if you are using Oracle Privileged Account Manager on IBM WebSphere.

17.2.1 Enabling TDE Mode

You can enable TDE mode by performing the following steps:

1. Section 17.2.1.1, "Enable TDE in the Database"
2. Section 17.2.1.2, "Enable Encryption in the Oracle Privileged Account Manager Schema"
3. Section 17.2.1.3, "Enable TDE Mode in the Oracle Privileged Account Manager Server Configuration"

17.2.1.1 Enable TDE in the Database

Enable TDE in the database by referring to one of the following procedures, depending on the Oracle Database version you are using:

- To enable TDE in Oracle Database Release 11.2, refer to the "Enabling Transparent Data Encryption" section in the *Oracle Database Advanced Security Administrator's Guide*.
- To enable TDE in Oracle Database Release 12.1, refer to the "Configuring Transparent Data Encryption" section in the *Oracle Database Advanced Security Administrator's Guide*.

Note:

- For additional information about storing data, refer to "Securing Stored Data Using Transparent Data Encryption" in the *Oracle Database Advanced Security Administrator's Guide*.
- Enabling TDE involves the creation of an encryption wallet. The wallet is a critical component of TDE, and should always be backed up. If the wallet is lost, encrypted data cannot be recovered. Refer to Oracle's best practices for "Transparent Data Encryption" at the following location:

  `http://www.oracle.com/technetwork/database/security/twp-transparent-data-encryption-bes-130696.pdf`

17.2.1.2 Enable Encryption in the Oracle Privileged Account Manager Schema

You can enable encryption in the Oracle Privileged Account Manager schema by using sqlplus (or any other client) to run the following `opamxencrypt.sql` script with the Oracle Privileged Account Manager schema user:

IAM_HOME`/opam/sql/opamxencrypt.sql`

For example,

```
sqlplus DEV_OPAM/password1 @IAM_HOME/opam/sql/opamxencrypt.sql
```

17.2.1.3 Enable TDE Mode in the Oracle Privileged Account Manager Server Configuration

You can enable TDE mode in the Oracle Privileged Account Manager server configuration by using one of the following methods:

- From the Oracle Privileged Account Manager Console
- From the Oracle Privileged Account Manager Command Line Tool

From the Oracle Privileged Account Manager Console

To enable TDE mode by using the Console, refer to step 3 in Section 5.2.3.1, "From the Console."

From the Oracle Privileged Account Manager Command Line Tool

To enable TDE mode (if the `tdemode` flag is set to `false`) by using the command line tool, complete the following steps:

Note: Before you begin, ensure that the Oracle Privileged Account Manager server is running.

1. Set the environment variables, *ORACLE_HOME* and *JAVA_HOME*.
2. Run the following script:

   On **UNIX**, type:

   ```
   ORACLE_HOME/bin/opam.sh -url OPAM_Server_Url -x modifyconfig -configtype global
   -propertyname tdemode -propertyvalue true -u OPAM_APPLICATION_CONFIGURATOR_USER
   -p Password
   ```

   On **Windows**, type:

   ```
   ORACLE_HOME\bin\opam.bat -url OPAM_Server_Url -x modifyconfig
   -configtype global -propertyname tdemode -propertyvalue true -u
   OPAM_APPLICATION_CONFIGURATOR_USER
   -p Password
   ```

3. Perform the steps described in the "Optional: Enabling TDE in Oracle Privileged Account Manager Data Store" section of the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*.

17.2.2 Disabling TDE Mode

You can switch to non-TDE mode by performing the following steps:

1. Section 17.2.2.1, "Disable TDE Mode in the Oracle Privileged Account Manager Server Configuration"
2. Section 17.2.2.2, "Disable Encryption in the Oracle Privileged Account Manager Schema"

17.2.2.1 Disable TDE Mode in the Oracle Privileged Account Manager Server Configuration

You can disable TDE mode in the Oracle Privileged Account Manager server by using one of the following methods:

- From the Oracle Privileged Account Manager Console
- From the Oracle Privileged Account Manager Command Line Tool

From the Oracle Privileged Account Manager Console

To disable TDE mode by using the Console, refer to step 3 in Section 5.2.3.1, "From the Console."

From the Oracle Privileged Account Manager Command Line Tool

To disable TDE mode by using the command line tool, complete the following steps:

Note: Before you begin, ensure that the Oracle Privileged Account Manager server is running.

1. Set the environment variables, *ORACLE_HOME* and *JAVA_HOME*.
2. Run the following script:

On **UNIX**:

```
ORACLE_HOME/opam/bin/opam.sh -url OPAM_Server_Url -x modifyconfig
-configtype global -propertyname tdemode -propertyvalue false
-u OPAM_APPLICATION_CONFIGURATOR_USER -p Password
```

Where ***OPAM_Server_Url*** is of the form:

```
https://OPAM_Managed_Server_Hostname:OPAM_Managed_Server_SSL_port/opam
```

On **Windows**:

```
ORACLE_HOME\opam\bin\opam.bat -url OPAM_Server_Url -x modifyconfig
-configtype global -propertyname tdemode -propertyvalue false
-u OPAM_APPLICATION_CONFIGURATOR_USER -p Password
```

Where ***OPAM_Server_Url*** is of the form:

```
https://OPAM_Managed_Server_Hostname:OPAM_Managed_Server_SSL_port/opam
```

17.2.2.2 Disable Encryption in the Oracle Privileged Account Manager Schema

You can disable encryption in the Oracle Privileged Account Manager schema by using sqlplus (or any other client) to run the following `opamxunencrypt.sql` script with the Oracle Privileged Account Manager schema user:

```
IAM_HOME/opam/sql/opamxunencrypt.sql
```

For example,

```
sqlplus DEV_OPAM/password1 @MW_HOME/Oracle_IDM1/opam/sql/opamxunencrypt.sql
```

17.3 Adding New Connectors to an Existing Oracle Privileged Account Manager Installation

This section describes the processes for adding new connectors to your existing Oracle Privileged Account Manager installation. The topics include:

- Section 17.3.1, "Adding Connectors Supplied by Oracle"
- Section 17.3.2, "Adding Custom Connectors"

17.3.1 Adding Connectors Supplied by Oracle

If you are adding new ICF connectors that are supplied by Oracle, then they will be accompanied by installation instructions. These instructions describe where to store the connector bundle and how to modify the installation specific `opam-config.xml` file.

17.3.2 Adding Custom Connectors

Oracle Privileged Account Manager can use custom connectors that you created or that were created by a third party. However, these connectors must strictly adhere to the ICF standard. After verifying that the connector is ICF-compliant, perform the following steps to deploy the connector for Oracle Privileged Account Manager consumption:

1. Put the connector bundle in a location on the file system where the bundle can be read by the Oracle Privileged Account Manager at run time.

2. Perform the following steps to create a configuration block for the connector and include that block in the installation specific `opam-config.xml` file:
   a. Design and create a relevant configuration block.

      Both the `opam-config.xml` and `opam-config.xsd` files contain documentation and an example at the beginning of the file describing how to create a configuration block.

   b. Ensure that this connector configuration block includes the file system location you specified for the connector bundle in step 1.

   c. Add the new connector configuration block to the `opam-config.xml` file by containing it in a `<connectorConfig>` block.

   d. Validate the modified `opam-config.xml` file against the `opam-config.xsd` file to ensure that the Oracle Privileged Account Manager server can read the modified file. You can use your favorite XML schema validation tool for this purpose.
3. Restart the Oracle Privileged Account Manager server.
4. Connect to Oracle Privileged Account Manager, and then add and configure a new target system using the newly added connector type.

17.4 Copying Passwords to the Clipboard

> **See Also:** Section 20.4, "Frequently Asked Questions" for more information about ZeroClipboard and this feature

Oracle Privileged Account Manager enables an end user to copy any checked out password directly to the clipboard without revealing the password in plain text. This ability provides greater security and eliminates the need to manually type the password to checkout privileged accounts.

This section discusses the following topics:

- Section 17.4.1, "Downloading and Deploying the ZeroClipboard Library Files on the Server"
- Section 17.4.2, "Installing the Adobe Flash Plug-in"

17.4.1 Downloading and Deploying the ZeroClipboard Library Files on the Server

Before you can copy the checked out passwords to the clipboard, you must deploy the ZeroClipboard library on the server.

> **Note:** For more information about ZeroClipboard, refer to:
>
> `https://github.com/zeroclipboard/ZeroClipboard`

You must download and deploy the ZeroClipboard v1.x. library from the following URL:

`https://github.com/zeroclipboard/zeroclipboard/tree/1.x-master`

To do so:

1. Click the **Download ZIP** button on the lower-right side of the page and save the ZIP file to a desired location.

2. Extract the contents of the Zip file and locate the **zeroclipboard-master** directory.
3. Deploy the ZeroClipboard library files on the web application server such as WebLogic, WebSphere, and so on.

   The following example explains the procedure to deploy ZeroClipboard library files on the WebLogic server:

   a. Create a directory named **ZeroClipboard** in Oracle Privileged Account Manager, for clipboard operations.

      For example: `$ORACLE_HOME/ZeroClipboard`

      Note: The name of the directory created in this step *must* be "ZeroClipboard" to allow Oracle Privileged Account Manager to successfully load files from this directory.

   b. Copy the ZeroClipboard.js and ZeroClipboard.swf files from the zeroclipboard-master directory to the $ORACLE_HOME/ZeroClipboard directory that you created in Step 3a of this procedure. Within the ZeroClipboard directory, create another directory named WEB-INF, such as, `$ORACLE_HOME/ZeroClipboard/WEB-INF`

   c. In the WEB-INF directory, create a file named web.xml with the following content:

      ```
      <?xml version="1.0" encoding="UTF-8"?>
      <!DOCTYPE web-app PUBLIC
      "-//Sun Microsystems, Inc.//DTD Web application 2.3//EN"
      http://java.sun.com/j2ee/dtds/web-app_2_3.dtd>
      <web-app>
      </web-app>
      ```

      The location of this .xml file must be similar to the following sample location:

      `$ORACLE_HOME/ZeroClipboard/WEB-INF/web.xml`

   d. Save the web.xml file.

   e. Log in to the WebLogic Server Administration console.

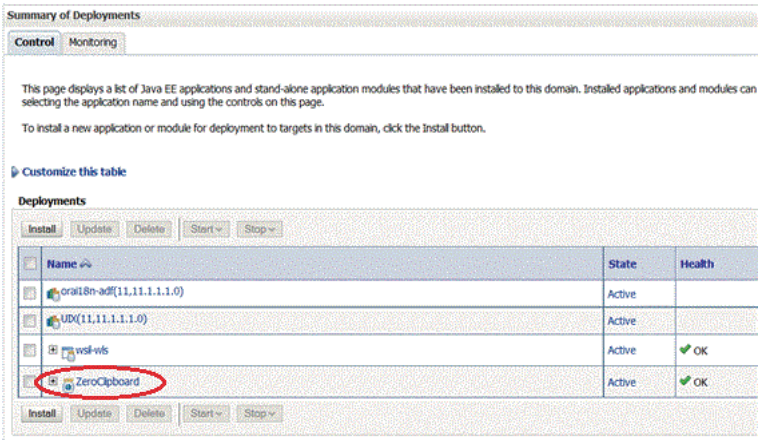
   f. Click **Deployments** (in the left pane) and **Install,** and specify the path to the ZeroClipboard directory you created in Step a.

   g. From the listed options, select **ZeroClipboard** and click **Next.**

   h. Retain the default options in "Install this deployment as an application" and click **Next.**

   i. Select the managed servers that are running Oracle Privileged Account Manager, and click **Finish.**

   j. The Deployment Success message will pop-up and you should be able to see ZeroClipboard in the Deployments screen as shown in the screenshot below:



17.4.2 Installing the Adobe Flash Plug-in

The Adobe Flash Player plug-in is used to copy passwords to clipboard in Oracle Privileged Account Manager. You must enable the Flash plug-in, download it, and install the latest version of this plug-in on your browser.

> **Note:** The Flash plug-in installation is required only on the web browser client is used to view the Oracle Privileged Account Manager Console. It is not required on the managed server that is running the application.

To download the Flash plug-in and ensure that you have correctly configured the plug-in, refer to:

`http://helpx.adobe.com/flash-player.html`

17.5 Advanced Management of Session Manager Data

This section describes how to manage your Oracle Privileged Session Manager (Session Manager) data. The topics include:

- Section 17.5.1, "Overview"
- Section 17.5.2, "Partitioning"
- Section 17.5.3, "Partition OPSM_SESSIONS Table"
- Section 17.5.4, "Purging"
- Section 17.5.5, "Managing Oracle Text Index for Session Recordings"

17.5.1 Overview

The Session Manager stores all of its session recording data in the Oracle Privileged Account Manager database schema. Over time, as more information is recorded, the disk footprint for this database schema will grow. Therefore, having a strategy to effectively manage this data is important.

Compliance regulations may require that you store audit data (such as session recordings) for long periods. You need a good backup and recovery plan to protect the data.

A good backup plan accounts for these basic guidelines:

- **Growth rate of session recordings**: The growth rate depends on the number of sessions and the type of activity (which results in recordable data) that occurs on those sessions. The growth of the session recording data generated daily determines, in turn, how often you want to perform backups.
- **Compliance regulations**: Consult your organization's compliance regulations to determine how frequently backups are required and for how many years session recording storage is mandatory.
- **Online or offline data management**: Consult your organization's compliance regulations to determine how frequently backups are required and what portion of session recording data must be easily accessible.

Oracle Database uses Oracle Recovery Manager (RMAN) for backup and recovery. For details, refer to:

- http://www.oracle.com/technology/deploy/availability/htdocs/BR_Overview.htm
- http://www.oracle.com/technology/deploy/availability/htdocs/rman_overview.htm

Note: The Oracle Privileged Account Manager schema is created using the Oracle Repository Creation Utility (RCU) and the session recording data in the Oracle Privileged Account Manager schema is stored in the `OPSM_SESSIONS` table.

17.5.2 Partitioning

The Oracle Privileged Account Manager schema is unpartitioned by default. However, session recording data is cumulative and older data is never removed. If you store a high volume of session recording data, then you should consider partitioning the `OPSM_SESSIONS` table, which allows for easier archiving.

Benefits of partitioning include:

- **Improved Performance**: If a table is range-partitioned by Timestamps, for example, queries by Timestamps can be processed on the partitions within that time-frame only.
- **Better Manageability**: You can create partitions on separate tablespaces (thus different disks), which enables you to move older data to slower and larger disks, while keeping newer data in faster and smaller disks.

  In addition, partitioning makes archiving much easier. For example, you can compress a single partition rather than having to partition the entire table.
- **Increased Availability**: If a single partition is unavailable, for example, and you know that your query can eliminate this partition from consideration, then the query can be successfully processed without needing to wait for the unavailable partition.

17.5.3 Partition OPSM_SESSIONS Table

In this example, the `OPSM_SESSIONS` table is partitioned on a quarterly basis. Depending on your needs, you can choose to implement a different partitioning scheme.

To minimize application down time, Oracle recommends that partitioning is done before using this schema for an Oracle Privileged Account Manager deployment. If

you are partitioning on an active Oracle Privileged Account Manager deployment, then you must first shut down all Oracle Privileged Account Manager processes before proceeding with the following steps.

The partitioning steps are as follows:

1. Login to the database using SQLPlus as the Oracle Privileged Account Manager schema user.
2. Rename the existing unpartitioned table. For example:

```
RENAME OPSM_SESSIONS TO OPSM_SESSIONS_NONPART;
```

3. Create a new partitioned table that follows the table structure of the unpartitioned table. This example uses the range-partitioning (by Timestamp) scheme:

```
CREATE TABLE OPSM_SESSIONS
PARTITION BY RANGE (STARTTIME)
(
    PARTITION OPSM_SESSIONS_DEFAULT VALUES LESS THAN (MAXVALUE)
)
AS SELECT * FROM OPSM_SESSIONS_NONPART;
```

4. Enable row movement to allow data to automatically move from partition to partition when new partitions are created. For example:

```
ALTER TABLE OPSM_SESSIONS ENABLE ROW MOVEMENT;
```

5. You can now create partitions. In this example, partitions are created by calendar quarter:

```
ALTER TABLE OPSM_SESSIONS
SPLIT PARTITION OPSM_SESSIONS_DEFAULT AT (TO_DATE('01/04/2013', 'DD/MM/YYYY'))
INTO (PARTITION OPSM_SESSIONS_Q1_2013, PARTITION OPSM_SESSIONS_DEFAULT)
UPDATE INDEXES;

ALTER TABLE OPSM_SESSIONS
SPLIT PARTITION OPSM_SESSIONS_DEFAULT AT (TO_DATE('01/07/2013', 'DD/MM/YYYY'))
INTO (PARTITION OPSM_SESSIONS_Q2_2013, PARTITION OPSM_SESSIONS_DEFAULT)
UPDATE INDEXES;

ALTER TABLE OPSM_SESSIONS
SPLIT PARTITION OPSM_SESSIONS_DEFAULT AT (TO_DATE('01/10/2013', 'DD/MM/YYYY'))
INTO (PARTITION OPSM_SESSIONS_Q3_2013, PARTITION OPSM_SESSIONS_DEFAULT)
UPDATE INDEXES;

ALTER TABLE OPSM_SESSIONS
SPLIT PARTITION OPSM_SESSIONS_DEFAULT AT (TO_DATE('01/01/2014', 'DD/MM/YYYY'))
INTO (PARTITION OPSM_SESSIONS_Q4_2013, PARTITION OPSM_SESSIONS_DEFAULT)
UPDATE INDEXES;
```

Note: You should periodically create new partitions for new quarters.

17.5.4 Purging

Purging removes the Oracle Privileged Account Manager session recording data from the Oracle Privileged Account Manager schema. Therefore, if you foresee needing to

revisit this data at a later point, then use Oracle Recovery Manager (RMAN) for backup and recovery.

Keep in mind that with a range-partitioned table it is much more efficient to drop a partition when you want to remove old data, rather than deleting individual rows.

```
ALTER TABLE OPSM_SESSIONS DROP PARTITION OPSM_SESSIONS_Q1_2013;
```

Once partitions are created, you can purge and back up a particular partition. Refer to the Oracle Database documentation for details.

17.5.5 Managing Oracle Text Index for Session Recordings

Oracle Privileged Account Manager uses Oracle Text CONTEXT Index to index session recordings. These indexes (OPSM_SESSIONS_METADATA and OPSM_SESSION_RMETADTA) are used when performing a pattern search for Checkout History.

To maintain these indexes perform the following:

1. Connect to the database using OPAM schema.
2. Find OPAM_BINSTORE_TS_NAME name using the following query:

   SELECT UNIQUE TABLESPACE_NAME FROM USER_SEGMENTS where tablespace_name LIKE'%_BINSTORE';

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.

```
BEGIN
  ctx_ddl.create_preference('opam_text_storage', 'BASIC_STORAGE');

  ctx_ddl.set_attribute('opam_text_storage','I_TABLE_CLAUSE', 'tablespace
 <OPAM_BINSTORE_TS_NAME> storage (initial 10M next 10M)');

   ctx_ddl.set_attribute('opam_text_storage', 'I_INDEX_CLAUSE', 'tablespace
 <OPAM_BINSTORE_TS_NAME> storage (initial 1M) compress 2');

   ctx_ddl.set_attribute('opam_text_storage', 'K_TABLE_CLAUSE', 'tablespace
 <OPAM_BINSTORE_TS_NAME> storage (initial 10M next 10M)');

   ctx_ddl.set_attribute('opam_text_storage', 'R_TABLE_CLAUSE', 'tablespace
 <OPAM_BINSTORE_TS_NAME> storage (initial 1M) lob (data) store as (cache)');

   ctx_ddl.set_attribute('opam_text_storage', 'N_TABLE_CLAUSE', 'tablespace
 <OPAM_BINSTORE_TS_NAME> storage (initial 1M)');

   ctx_ddl.set_attribute('opam_text_storage', 'P_TABLE_CLAUSE', 'tablespace
 <OPAM_BINSTORE_TS_NAME> storage (initial 1M)');
 END;
 /
```

4. Apply the new storage preference using the following commands:

   ALTER INDEX opam_text_storage REBUILD PARAMETERS;

   ALTER INDEX opam_text_storage' REBUILD PARAMETERS;

5. Verify that the above tables are moved to the new tablespace by querying USER_SEGMENTS table.
6. Make sure the Text index status is valid after this step.

17.5.5.1 Text Index Optimization

The Text index could become fragmented due to on-going operations on the Text index. Optimizing the text index on regular basis removes the old data and minimizes the fragmentations, which can improve the search performance.To perform this, Oracle Privileged Account Manager has introduced the following Oracle Database scheduler jobs:

- FAST_OPTIMIZE_METADATA
- REBUILD_OPTIMIZE_RMETADATA
- FAST_OPTIMIZE_RMETADATA
- REBUILD_OPTIMIZE_METADATA

These jobs reside in OPAM 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_METADATA and FAST_OPTIMIZE_RMETADATA meant to be running on frequent basis. By default, it is scheduled to run once a day at 1 AM and 2 AM respectively.

REBUILD_OPTIMIZE_METADATA and REBUILD_OPTIMIZE_RMETADATA does a full optimization and rebuilds the Text index. They are not meant to be running frequent basis. By default, REBUILD_OPTIMIZE_METADATA and REBUILD_OPTIMIZE_RMETADATA is scheduled to run every Sunday at 2 AM and 4 AM respectively. 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.

Make sure the default schedule (daily 1 AM and 2 AM for FAST and every Sunday 2 AM and 4 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.

Enable the jobs using the following commands:

```
.
    BEGIN
    DBMS_SCHEDULER.ENABLE ('FAST_OPTIMIZE_METADATA');
    END;
    /
   BEGIN
   DBMS_SCHEDULER.run_job ('REBUILD_OPTIMIZE_METADATA');
   END;
   /
.
```

> **Note:** The Text index optimization can be done when the server is up and search is taking place.

17.5.5.2 Updating the Synchronization Frequency

By default, the index is synchronized every hour. You can update the synchronization frequency by using sqlplus (or any other client) to run the following opamupdateotextind.sql script with the Oracle Privileged Account Manager schema user:

```
IAM_HOME/opam/sql/opamupdateotextind.sql
```

For example,

```
sqlplus DEV_OPAM/pwd1 @IAM_HOME/opam/sql/opamupdateotextind.sql
```

You must enter the new frequency (in minutes) when prompted.

For more information about Oracle Text CONTEXT Index, refer to "Indexing with Oracle Text" in *Oracle Text Application Developer's Guide.*

17.6 Moving from a Test Environment to a Production Environment

For information about moving Oracle Fusion Middleware components from one environment to another, refer to "Moving from a Test to a Production Environment" in *Oracle Fusion Middleware Administrator's Guide*.

For information about moving Identity Management components, including Oracle Privileged Account Manager, from a test environment to a production environment, refer to "Moving Identity Management Components to a Target Environment" in *Oracle Fusion Middleware Administrator's Guide*.

17.7 Rebranding Oracle Privileged Account Manager

If necessary, you can rebrand the Login and Oracle Privileged Account Manager pages. The following topics contain instructions for changing the page title, branding text, and logo image on these pages:

- Section 17.7.1, "Customizing the Login Page"
- Section 17.7.2, "Customizing the Oracle Privileged Account Manager Page"
- Section 17.7.3, "Customizing the About Oracle Information"

Tip: Create a back-up copy before you modify any files.

17.7.1 Customizing the Login Page

You configure branding changes for the Login page in the `oinav.ear/oiNavApp-war.war/SignIn.jspx` file.

Login Page Title

To change the Login page title, modify the title in **`af:document "#{signinBean.signInTitle}"`**.

Refer to the following code sample:

```
<af:document id="d1" title="#{signinBean.signInTitle}" theme="dark"
 initialFocusId="pt1:_pt_it1">
```

Login Page Branding Text

To change the branding text on the Login page, modify the value of **`af:outputText "#{signinBean.title}"`**, which is defined in the branding facet.

Refer to the following code sample:

```
<f:facet name="branding">
  <af:outputText value="#{signinBean.title}" id="ot1"/>
</f:facet>
```

Login Page Logo Image

To change the logo image on the Login page, perform these steps:

1. Copy the new image, for example `newlogo.png`, into the following directory:

   `oinav.ear/oiNavApp-war.war/images`

2. To skip the default logo, add the following line to the `oinav.ear/oiNavApp-war.war/SignIn.jspx` file:

   ```
   <f:attribute name="brandingLogoCls" value=""/>
   ```

3. If the new logo's image size is larger than the default size 30, add the following line to adjust the header size:

   ```
   <f:attribute name="globalBrandingSize" value="60"/>
   ```

4. Modify the branding facet by replacing **`newlogo.png`**, **`newlogo mouse over text`**, and **`new branding text`**.

   Refer to the following code sample:

   ```
   <f:facet name="branding">
     <af:panelGroupLayout layout="horizontal">
       <af:image source="/images/newlogo.png" shortDesc="newlogo mouse over text"
   id="im1"/>
       <af:spacer width="5"/>
       <af:outputText value="new branding text" id="ot1"/>
     </af:panelGroupLayout>
   </f:facet>
   ```

17.7.2 Customizing the Oracle Privileged Account Manager Page

You configure branding changes for the Oracle Privileged Account Manager page in the `oinav.ear/oiNavApp-war.war/opam.jspx` file.

Oracle Privileged Account Manager Page Title

To change the page title on the Oracle Privileged Account Manager page, modify the title in **`af:document "#{resBundle.PRODUCT_OPAM}"`**

Refer to the following code sample:

```
<af:document title="#{resBundle.PRODUCT_OPAM}" id="d1" theme="contentBody">
```

Oracle Privileged Account Manager Branding Text

To change the branding text on the Oracle Privileged Account Manager page, modify the value of **`af:outputText "#{resBundle.OPAM_PRODUCT_TITLE}"`**, which is defined in the branding facet.

Refer to the following code sample:

```
<f:facet name="branding">
  <af:outputText value="#{resBundle.OPAM_PRODUCT_TITLE}" id="ot1"/>
</f:facet>
```

Oracle Privileged Account Manager Page Logo Image

To change the logo image on the Oracle Privileged Account Manager page, perform these steps:

1. Copy the new image, for example `newlogo.png`, into the following directory:

   `oinav.ear/oiNavApp-war.war/images`

2. To skip the default logo, add the following line to the `oinav.ear/oiNavApp-war.war/opam.jspx` file:

```
<f:attribute name="brandingLogoCls" value=""/>
```

3. If the new logo's image size is larger than the default size 30, add the following line to adjust the header size:

```
<f:attribute name="globalHeaderSize" value="30"/>
```

4. Modify the branding facet by replacing **`newlogo.png`**, **`newlogo mouse over text`**, and **`new branding text`**.

   Refer to the following code sample:

```
<f:facet name="branding">
  <af:panelGroupLayout layout="horizontal">
    <af:image source="/images/newlogo.png" shortDesc="newlogo mouse over text"
id="im1"/>
    <af:spacer width="5"/>
    <af:outputText value="new branding text" id="ot1"/>
  </af:panelGroupLayout>
</f:facet>
```

17.7.3 Customizing the *About Oracle* Information

This section describes how to hide or replace the About Oracle information.

To Hide the About Oracle Link:

Add the following line in the `oinav.ear/oiNavApp-war.war/SignIn.jspx` file:

```
<f:attribute name="manageGlobalNav" value="true"/>
```

To Replace the About Oracle Link and Text:

1. Extract the contents of the archive from the following location to a temporary folder.

   ORACLE_HOME`/modules/oracle.idm.uishell_11.1.1/oracle.idm.uishell.war`

2. Extract the contents of the `/WEB-INF/lib/oracle-idm-uishell.jar` file to a temporary location.

3. Search for **`ABOUT_ORACLE`** in the `templates/IdmSignIn.jspx` files, and replace the text and destination attribute:

```
<af:commandNavigationItem text="#{resBundle.ABOUT_ORACLE
destination="http://www.oracle.com/us/corporate/index.htm" id="_pt_cni1"/>
```

4. Repackage the JAR file and put it in the `/WEB-INF/lib/` directory of the extracted Web Archive (WAR) file.

5. Repackage the `oracle.idm.uishell.war` file and put it back in its original location.

18

Developing Plug-Ins for Oracle Privileged Account Manager

This chapter describes how to develop your own plug-ins for Oracle Privileged Account Manager.

> **Note:** For basic information about managing Oracle Privileged Account Manager plug-ins, including how to configure and deploy plug-ins, refer to Section 13, "Working with Plug-Ins."

This chapter includes the following sections:

- Section 18.1, "Overview"
- Section 18.2, "Setting Up a Plug-In"
- Section 18.3, "Understanding the Plug-In API"
- Section 18.4, "Debugging and Logging for Plug-Ins"
- Section 18.5, "Example Plug-ins"
- Section 18.6, "Managing Plug-Ins"

18.1 Overview

You can use Oracle Privileged Account Manager's Java-based plug-in framework to create plug-ins that extend Oracle Privileged Account Manager's functionality to accommodate your specific business and technical requirements. Plug-ins enable you to provide custom logic within a transaction or to connect to a custom data source.

The topics in this section include:

- Section 18.1.1, "Oracle Privileged Account Manager Framework Packages"
- Section 18.1.2, "Special Considerations for Using Oracle Privileged Account Manager Plug-Ins"

18.1.1 Oracle Privileged Account Manager Framework Packages

The Oracle Privileged Account Manager plug-in framework contains the plug-in interfaces and classes you need to develop a plug-in implementation. This framework is shipped in the following jar file:

ORACLE_HOME/opam/jlib/opam-plugin-framework.jar

You can use this `jar` file to develop, implement, and compile plug-ins.

> **Note:** Refer to Section 13.2, "Developing Plug-Ins for Oracle Privileged Account Manager" for additional information about developing plug-ins for Oracle Privileged Account Manager.

18.1.2 Special Considerations for Using Oracle Privileged Account Manager Plug-Ins

Following are some special considerations and dependencies that you must consider when developing plug-ins:

- Although an Oracle Privileged Account Manager server runtime is not required when developing plug-ins, the server runtime is required for deployment and testing purposes.

18.2 Setting Up a Plug-In

To set-up an Oracle Privileged Account Manager Java plug-in,

1. Create a standalone `Java` program using the predefined interface and implement the required methods.

   You can execute the plug-in using *pre* or *post* timing for an operation. You must implement the corresponding pre plug-in or post plug-in interface.

   > **Note:** Refer to Section 13.2.7.1, "Pre-Operation Plug-Ins" and Section 13.2.7.2, "Post-Operation Plug-Ins" for a description of these timings.

2. Before compiling, place the following `jar` in your classpath:

   ORACLE_HOME`/opam/jlib/opam-plugin-framework.jar`

3. Compile the plug-in Java files and create the `class` or `jar` file. Ensure the compilation completes without errors.
4. Put the `class` or `jar` file in a file system location that is accessible to the Oracle Privileged Account Manager server. For high-availability cluster configurations, you may want to place the `class` or `jar` file in each individual node or in a shared location that is accessible to all nodes.
5. Register the plug-in by adding the plug-in configuration entry.

   > **Note:**
   >
   > - For information about managing plug-in configurations from the Console, refer to Section 13.3, "Creating a Plug-In Configuration."
   > - For information about managing plug-in configurations from the command line, refer to Section A.9, "Working with Plug-Ins."

You can choose any name and package for the `class` and `jar` files. However, you must be sure to use the same name and package when configuring the `Plug-in Class Name` and the `Plug-in Class Path` attributes when registering the plug-in.

For example, if you create a plug-in using a fully qualified name, such as *my.sample.OpamPlugin*, that is compiled to the `/u01/myplugin.jar` file and that has a

dependency on some classes in `/u01/myutils.jar`, then your plug-in configuration must use the following:

Plug-in Class Name	Plug-in Classpath
my.sample.OpamPlugin	`/u01/myplugin.jar` `/u01/myutils.jar`

After the plug-in configuration is registered and enabled, the server invokes the plug-in whenever the invocation criteria are met.

18.3 Understanding the Plug-In API

This section presents a high-level overview of the plug-in API and explains the role of the main classes and interfaces.

> **Note:** Do not use `System.exit()` in a plug-in implementation because it might cause failures in the server runtime.

The topics in this section include:

- Section 18.3.1, "Communication between the Server and Plug-In"
- Section 18.3.2, "Plug-In Structure"
- Section 18.3.3, "Plug-In Interfaces and Classes"

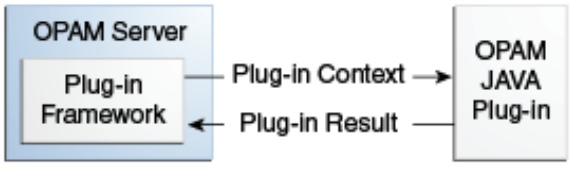
18.3.1 Communication between the Server and Plug-In

Oracle Privileged Account Manager plug-ins use the *PrePlugin* or *PostPlugin* interface to communicate with the Oracle Privileged Account Manager server. When invoking a plug-in, the server constructs the `PluginContext` object and passes details about the operation (such as target, account, and so forth) to the plug-in invoked by that operation. The server also passes the operation request body and the plug-in configuration that invokes the plug-in to the plug-in.

The plug-in constructs a `PluginResult` object. After completing its task, the plug-in passes the `PluginResult` object back to the server. The `PluginResult` object can contain a success or failure code, custom error messages, and log messages. In some cases, the plug-in can change or add details to the request body and pass those details back to the server.

The following figure illustrates how Oracle Privileged Account Manager plug-ins communicate with the server.

Figure 18–1 How Plug-Ins Communicate with the Server



18.3.2 Plug-In Structure

The general structure for a Java plug-in is as follows:

For a Pre Plug-In

```
public class OPAM_PLUGIN_CLASSNAME implements PrePlugin {
    public void runPrePlugin(PluginContext ctx, String reqBodyJSON, String
pluginCfgJSON) {
       // Plugin Code
   }
}
```

For a Post Plug-In

```
public class OPAM_PLUGIN_CLASSNAME implements PostPlugin {
    public void runPostPlugin(PluginContext ctx, String reqBodyJSON, String
pluginCfgJSON) {
       // Plugin Code
   }
}
```

18.3.3 Plug-In Interfaces and Classes

> **Note:** Refer to the *Oracle Privileged Account Manager Plug-In Framework Java API Reference* for more information.

This section describes the plug-in interfaces and classes to use for Oracle Privileged Account Manager.

The topics in this section include:

- Section 18.3.3.1, "PlugInContext"
- Section 18.3.3.2, "PluginResult"
- Section 18.3.3.3, "PrePlugin"
- Section 18.3.3.4, "PostPlugin"
- Section 18.3.3.5, "Property File"

18.3.3.1 PlugInContext

The Oracle Privileged Account Manager server creates the `PluginContext` object during plug-in invocation and sends it to the plug-in. This object contains the following information:

- **Account information in JSON format**. Obtained by using the `getAccountJSON()` method.
  - Account information is present in operations involving accounts, such as `checkin` and `checkout`.
  - Account information is not present in operations that do not involve accounts, such as adding or deleting a target. Also, account information is not present when you add a new account because the account is not yet created, and the details sent by the client can be obtained in the request body JSON.

> **Note:** Refer to Section B.7, "Account Resource" in Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface" for more information about the Account Resource and its JSON format.

- **Target information in JSON format**. Obtained by using the `getTargetJSON()` method.

  Similar to Account information in JSON format (as described in the preceding point), target information is only present in operations involving targets.

  > **Note:** Refer to Section B.6, "Target Resource" in Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface" for more information about the Target Resource and its JSON format.

- **Operation information in JSON format**. Obtained by using the `getOperationJSON()` method.

  This method contains information about the resource that this operation is being performed on, such as target, account, and so on. It also contains information about which operation is being performed, such as `add`, `delete`, and so on. For example:

  ```
  {
  "resourceType":"account",
  "operationName":"add"
  }
  ```

- **Authentication information in JSON format**. Obtained using the `getAuthContextJSON()` method.

  This method contains information about which user is performing the operation and in which groups that user is a member. For example

  ```
  {
  "requestor":"johndoe",
    "requestorGroups":["ITADMINS", "MANAGERS"]
  }
  ```

  This information is not present for operations of resource type `server` because the Oracle Privileged Account Manager server performs the operation rather than an end user.

- **PluginResult Object**. Must be created by the plug-in to pass information back to the server. Set this object by using the `setPluginResult()` method and obtain the value that was set by using `getPluginResult()` method.

  > **Note:** Refer to Section 18.3.3.2, "PluginResult" for additional information about the `PluginResult` object.

- **HTTP Response Status** (for *post* plug-ins only). Obtained using the `getHTTPResponseStatus()` method.

For post plug-ins, when the Oracle Privileged Account Manager server REST API-based operation has completed, the operation response code such as 200 OK, 401 Unauthorized, and so on, is sent to the plug-in.

- **HTTP Response Entity** (for *post* plug-ins only). Obtained using the `getHTTPResponseEntity()` method.

  Note: This entity is helpful if you have to write a custom plug-in to update the custom credential store, as this method returns the account name and the password which has been changed by Oracle Privileged Account Manager.

  For post plug-ins, when the Oracle Privileged Account Manager Server REST API-based operation has completed, the operation results are sent to the plug-in. For example, operation results might include a list of accounts in JSON format from a search operation, an account token in JSON format from a `checkout` operation, and so forth.

  The following is an example JSON for the HR_ADMIN account for the post timing on server for the "accountpasswordchange" event:

```
{
"accountToken":
{"accountName":"HR_ADMIN",
"accountUID":"ea75011d2ed48154a17d101cdb52c7b4",
"accountPassword":"password"
}
}
```

- **HTTP Response Location > Created GUID** (for *post* plug-ins only). Obtained using the `gettHTTPResponseLocation() -> getGuid()` method.

  For post plug-ins and `add` operations, when the Oracle Privileged Account Manager Server REST API-based `add` operation has completed, the newly created entity's GUID is sent to the plug-in. This GUID is only present for creation operations. The GUID is not present for operations such as delete target, modify target, and so forth.

18.3.3.2 PluginResult

The plug-in creates `PluginResult` and uses it to send information back to the Oracle Privileged Account Manager server.

To create this object, the plug-in uses the `PluginResult(java.lang.String resultJSON,int plgErrorCode,java.lang.String plgErrorMsg)` method. After creation, `PluginResult` is passed back to the server by storing it in the `PluginContext` sent by the server. For example

```
PluginResult result = new PluginResult(resultJSON, plgErrorCode, plgErrorMsg);
ctx.setPluginResult(result);
```

The `PluginResult` object contains the following information:

- **Plugin modified request body JSON**. Result information in JSON format.
  - In *pre* plug-in operations, the plug-in uses this information to change or add details to the request body JSON that is passed to it and then returns the updated content as result JSON. The server uses the result JSON to replace the request body JSON before performing the operation.

If multiple pre plug-ins are executed for the same operation, then each plug-in gets the updated request body based on their order of execution. After the last pre plug-in is executed, the final request body is used to execute the operation in the server.

If the plug-in does not want to modify the request body JSON, it can be passed as is into the plug-in result JSON. The plug-in can set this value when creating the `PluginResult` object and can access it by using the `getResultJSON()` and `setResultJSON()` methods.

– In *post* plug-in operations, the plug-in can read this value but modifying the value has no impact because the operation has already been completed.

- **Error code**. Specifies success (`PluginResult.CODE_SUCCESS`) or failure (`PluginResult.CODE_FAILURE`).

  The plug-in can set this value during the `PluginResult` object creation and can access it by using the `getErrorCode()` and `setErrorCode()` methods.

- **Error message**. Describes the error in case of failures.

  The plug-in can set any string message and pass it back to the client. The plug-in can set the message during the `PluginResult` object creation and can access it by using the `getPlgErrorMsg()` and `setErrorMs()` methods.

  For example, assume you configured a pre plug-in for `checkout` operations to validate checkout dates and prevent account usage on blacklisted dates, such as regional holidays, weekends, etc. The plug-in can set an error message on checkout such as, `"Checkout is not allowed on holidays."` In this case, the `checkout` operation will fail and it will include the error message, `"Plug-in execution failed with error message: Checkout is not allowed on holidays."`

- **Debug logging**. Provides information about the plug-in execution.

  The plug-in can pass log messages back to the server that will be logged in the server log file. You can use these logs for debugging the plug-in execution. access the log message by using the `getLog()`, `appendLog(String)`, and `clearLog()` methods.

18.3.3.3 PrePlugin

You must implement the PrePlugin interface to create a pre plug-in for Oracle Privileged Account Manager operations. Use following method to implement the PrePlugin interface:

```
void runPrePlugin(PluginContext ctx, java.lang.String reqBodyJSON,
java.lang.String pluginCfgJSON)
```

The following information is passed from the server to the pre plug-in through `runPrePlugin` method:

- **PluginContext object**. Contains details about the operation.

  Refer to Section 18.3.3.1, "PlugInContext" for more information.

- **Request Body JSON**. Provides the request sent by the client to Oracle Privileged Account Manager for the REST API operation.

  The plug-in can modify this information for pre plug-ins and it can send the newly updated JSON back to the server in the `PluginResult` object. Refer to Section 18.3.3.2, "PluginResult" for more information.

- **Plug-in configuration**. Invokes this operation in JSON format.

  You can use the same Java plug-in implementation for many operations. For example, you can use the same plug-in that performs email notification for both `checkout` and `checkin` operations.

  When the plug-in configuration is passed to the plug-in, it provides details to the plug-in on which the configuration is causing this invocation, which is also useful for passing custom attributes that are present in the configuration to the plug-in. For example, in the email notification case, you can store mail server details as custom attributes in the plug-in configuration and they will be passed to the plug-in.

18.3.3.4 PostPlugin

You must implement the PostPlugin interface to create a post plug-in for Oracle Privileged Account Manager operations. Use following method to implement the PostPlugin interface:

```
void runPostPlugin(PluginContext ctx, java.lang.String reqBodyJSON,
java.lang.String pluginCfgJSON)
```

The information that is passed from the server to the post plug-in through the `runPostPlugin` method is the same as the information described in the previous section. Review the list in Section 18.3.3.3, "PrePlugin."

18.3.3.5 Property File

Plug-ins support custom attributes that are used to pass input parameters from the plug-in configuration to the plug-in implementation. As these attribute names are defined by the plug-in developer, they must be manually configured along with their values by the administrator. To simplify the process, the property file is used to define the attribute names and values.

The developer can create a .properties file named after the defined plug-in and package it in the .jar file. For example, if the plugin is named "my.sample.OpamPlugin", then the .properties file is named as "my.sample.OpamPlugin.properties". The properties file should be packaged in the root folder of the compiled plug-in .jar file.

Within the .properties file, a list of custom attributes and the following information can be defined for each plug-in:

- The default value or values
- If the configuration of a custom attribute is mandatory
- If the value of a custom attribute should be masked

The administrator can load a .properties file while configuring a plug-in. If a .properties file exists for a plug-in, then:

- The default value of custom attributes can be loaded to the plug-in configuration using the Load button in the Console. Refer to Step 5 of Section 13.3, "Creating a Plug-In Configuration" for detailed information.
- The mandatory plug-in custom attributes will be checked as defined in the .properties file.
- Custom attribute values will be masked as defined in the .properties file.

The following is the snippet of a sample .properties file named examplePlugin.properties within the examplePlugin.jar:

Note: The `name`, `default`, `mask`, and `required` properties that are defined in the following snippet are mandatory. If you want to supply a blank value for these properties, you must configure the value as `=""`

```
<customattribute>
    <name value="email"/>
    <default value="abc@efg.com"/>
    <mask value="false"/>
    <required value="true"/>
  </customattribute>
  <customattribute>
    <name value="day"/>
    <default value=["Saturday,Sunday"]/>
    <mask value="false"/>
    <required value="false"/>
  </customattribute>
```

18.4 Debugging and Logging for Plug-Ins

A plug-in can maintain its own log file and can log to that file in real time. In addition, a plug-in can log debug messages in the Oracle Privileged Account Manager server log file during execution by using `PluginResult` object debug logging methods, as described in Section 18.3.3.2, "PluginResult."

Messages logged using the `PluginResult` method will be present in the Oracle Privileged Account Manager Server log file. To view these messages, you must set the logging level to `TRACE:32` (very detailed trace or debug information).

The following example shows the sample code used to implement a plug-in's logging functionality.

Example 18–1 Sample Code Used to Implement Plug-In Logging

```
public void runPostPlugin(PluginContext ctx, String reqBodyJSON,
String pluginCfgJSON) { // the parameters are the same for runPrePlugin()
…
  PluginResult result = ctx.getPluginResult(); // get the PluginResult object
    from PluginContext object
  result.appendLog("Here is the log");  // append log
…
  // System.out.println(result.getLog()); // getLog() will return current log
  // result.clearLog(); // clearLog() will remove the log that has been recorded
…
}
```

18.5 Example Plug-ins

This section discusses the following examples:

- Section 18.5.1, "Pre Plug-In Example"
- Section 18.5.2, "Post Plug-In Example"

18.5.1 Pre Plug-In Example

Example 18–3 illustrates a pre plug-in that blocks an operation, based on the specified dates, before the operation is executed.

Your organization may have some blacklist dates, such as regional holidays and yearly closures, when access to privileged accounts should not be allowed. This pre plug-in performs the validation and extends Oracle Privileged Account Manager's Usage Policy functionality.

This section discusses the following topics:

- Section 18.5.1.1, "Configuring a Pre Plug-In"
- Section 18.5.1.2, "Compiling a Pre Plug-In"

18.5.1.1 Configuring a Pre Plug-In

You can define the attributes of a plug-in to configure it to perform specific actions. Use the attributes described in the following table to configure the plug-in:

Attribute Name	Attribute Value
pluginName	BlackListDates
pluginStatus	active
pluginResource	account
pluginOperation	checkout
pluginTiming	pre
pluginOrder	1
pluginClassName	BlackListDates
pluginClassPath	`/myhome/plugins/BlackListDates.jar`
pluginCustomAttrs	Use the **`date`** custom attribute to specify one or more blacklist dates. For example: ■ **date**: 10/01 ■ **date**: 06/30
pluginTimeout	60

This plug-in uses the following custom attributes to specify the blacklist dates.

Attribute Name	Attribute Value
date	Month/Day, for example 10/01. **Note**: ■ You can specify more than one `date` attribute. ■ Do not ignore the zeros (0) used in the plug-in sample. For example, to block June 9, you must specify **`06/09`** instead of **`6/9`**.

Optionally, perform the procedure in this section to configure custom attributes for a pre-plugin using a property file. If a property file is defined, then the load button can be used to load the properties while creating the plugin configuration. If the property file is not defined, then you can manually type the custom attributes while creating the plugin configuration.

To configure custom attributes using a property file:

1. Create a file named "BlackListDates.properties" in the same folder that contains the "BlackListDates.class" file.
2. Define a default value for the "date" attribute.
3. Save the .properties file.

Example 18–2, "Custom Attributes of the Blacklist Dates Pre Plug-In Property File" provides the sample content of this .properties file.

Example 18–2 Custom Attributes of the Blacklist Dates Pre Plug-In Property File

```
<customattribute>
   <name value="date"/>
   <default value=["10/01,12/01"]/>
   <mask value="false"/>
   <required value="true"/>
 </customattribute>
 <customattribute>
   <name value="version"/>
   <default value=["1.0.0"]/>
   <mask value="false"/>
   <required value="false"/>
 </customattribute>
```

18.5.1.2 Compiling a Pre Plug-In

After you configure a pre plug-in, you must compile it. Perform the procedure described in this section to do so:

1. If necessary, download the following files:

Download From	Files to Download
ORACLE_HOME	▪ opam-plugin-framework.jar ▪ jettison-1.3.jar ▪ jersey-bundle-1.18.jar ▪ jsr311-api.jar

2. Use the following sample compile command:

```
javac -cp .:ORACLE_HOME/opam/jlib/opam-plugin-framework.jar:
ORACLE_HOME/opam/jlib/third-party/jettison-1.3.jar:
ORACLE_HOME/opam/jlib/third-party/jersey-bundle-1.18.jar:
ORACLE_HOME/opam/jlib/third-party/jsr311-api.jar
BlackListDates.java
```

3. Create a jar file:

```
jar -cf BlackListDates.jar BlackListDates.class BlackListDates.properties
```

After you configure this plug-in for Oracle Privileged Account Manager, the plug-in will block the user from executing the Oracle Privileged Account Manager operation based on the `date` attributes. For example, if today is 10/01, and one of the date attributes is 10/01, then the user will not be able to perform this operation.

Note: In the following example, only the month and day are necessary. The year does not matter.

Example 18–3 Blacklist Dates Pre Plug-In

```
import org.codehaus.jettison.json.JSONException;
import org.codehaus.jettison.json.JSONObject;
import org.codehaus.jettison.json.JSONStringer;
import org.codehaus.jettison.json.JSONArray;

import com.oracle.idm.opam.plugin.interfaces.PrePlugin;
import com.oracle.idm.opam.plugin.context.PluginContext;
import com.oracle.idm.opam.plugin.context.PluginResult;

import java.util.Date;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;

public class BlackListDates implements PrePlugin {

public void runPrePlugin(PluginContext ctx, String reqBodyJSON, String pluginCfgJSON){
  System.out.println("===== In BlackListDates.runPrePlugin =====");
  try {
    JSONArray blackDates = null;
    JSONObject plugin = new JSONObject(pluginCfgJSON);
    JSONObject config = plugin.getJSONObject("plugin");
    JSONArray customAttrsArr = new JSONArray();
    if(config.has("pluginCustomAttrs"))
      customAttrsArr = config.getJSONArray("pluginCustomAttrs");
    for(int i=0; i<customAttrsArr.length(); i++) {
      JSONObject singleJSON = customAttrsArr.getJSONObject(i);
      JSONObject singleAttr = singleJSON.getJSONObject("pluginCustomAttr");
      String attrName = singleAttr.getString("attrname");
    if(attrName.equalsIgnoreCase("date"))
      blackDates = singleAttr.getJSONArray("attrvalue");
    }

    for(int i=0; i<blackDates.length(); i++) {
      String date = blackDates.getString(i);
      String[] count = date.split("/");
      if(count.length != 2)// wrong format, ignore
        continue;
      else {
        if(isToday(date)) {
          setResult(ctx, reqBodyJSON, PluginResult.CODE_FAILURE,
 "You are not allowed to do the operation on this date : " + date);
         return;
        }
      }
     }
   }
  catch (Exception e) {
    System.out.println("Exception happened: ");
    e.printStackTrace();
    PluginResult ret = new PluginResult(reqBodyJSON, PluginResult.CODE_FAILURE, e.getMessage());
    ctx.setPluginResult(ret);
  }
  System.out.println("===== Finished BlackListDates.runPrePlugin =====");
```

```
}
private static boolean isToday(String blackDate) {
  DateFormat dateFormat = new SimpleDateFormat("MM/dd");
  Date date = new Date();
  if( blackDate.equalsIgnoreCase( dateFormat.format(date).toString() ) )
    return true;
  else
    return false;
 }
private static void setResult(PluginContext ctx, String reqBodyJSON, int resultCode, String msg) {
  PluginResult ret = new PluginResult(reqBodyJSON, resultCode, msg);
  ctx.setPluginResult(ret);
  return;
}

}
```

18.5.2 Post Plug-In Example

Example 18–4 illustrates a post plug-in that sends an email about the operation after the operation has completed.

This section discusses the following topics:

- Section 18.5.2.1, "Configuring a Post Plug-In"
- Section 18.5.2.2, "Compiling a Post Plug-In"

18.5.2.1 Configuring a Post Plug-In

You can define the attributes of a plug-in to configure it to perform specific actions. Use the attributes described in the following table to configure the plug-in:

Attribute Name	Attribute Value
pluginName	EmailNotification
pluginDescription	This is an Email Notification Plug-in.
pluginResource	account
pluginOperation	checkout
pluginTiming	post
pluginOrder	1
pluginClassName	EmailNotificationPlugin **Note**: Class Name should be consistent with the class name in your plug-in jar file

pluginClassPath	`/myhome/plugins/EmailNotificationPlugin.jar`
	Note: The `pluginClassPath` is a multi-value attribute, so the JSON should be sent in JSON array format, similar to the following:
	`pluginClassPath: ["/myhome/plugins/EmailNotificationPlugin.jar"]`

Attribute Name	Attribute Value
pluginCustomAttrs	Use the following custom attributes to send information, such as the SMTP server to use for sending the email, the address where the email is sent, and so forth: ▪ **smtp_server**: SMTP host name ▪ **smtp_port**: 25 ▪ **to_addr**: to_address@somedomain ▪ **from_addr**: from_address@somedomain

You must also configure the following custom attributes to send information, such as the SMTP server to use for sending the email, the address where the email is sent, and so forth.

Attribute Name	Attribute Value
smtp_server	SMTP host name
smtp_port	25
to_addr	*<EMAIL ADDRESS>* **Note**: You can set up multiple values. Enter as many `to_addr` and values as required.
from_addr	*<EMAIL ADDRESS>*
user	*Optional*. SMTP login user name
password	*Optional*. SMTP login user password

This plug-in uses the JAVA Mail API to send emails. For more information about the JAVA Mail library and to download the library, refer to the following link:

`http://www.oracle.com/technetwork/java/javamail/index.html`

Optionally, perform the procedure in this section to configure custom attributes for a post-plugin using a property file. If a property file is defined, then the load button can be used to load the properties while creating the plugin configuration. If the property file is not defined, then you can manually type the custom attributes while creating the plugin configuration.

To configure custom attributes using a property file:

1. Create a file named "EmailNotificationPlugin.properties" in the same folder that contains the "EmailNotificationPlugin.class" file.
2. Define a default value for the "date" attribute.
3. Save the .properties file.

Example 18–4, "Email Notification Post Plug-In Property File" provides the sample content of this .properties file.

18.5.2.2 Compiling a Post Plug-In

After you configure a post plug-in, you must compile it. Perform the procedure described in this section to do so:

1. If necessary, download the following files:

Download From	Files to Download
ORACLE_HOME	■ opam-plugin-framework.jar ■ jettison-1.3.jar ■ jersey-bundle-1.18.jar ■ jsr311-api.jar
JAVA Mail API	mail.jar

2. Use the following sample compile command:

```
javac -cp .:ORACLE_HOME/opam/jlib/opam-plugin-framework.jar:
ORACLE_HOME/opam/jlib/third-party/jettison-1.3.jar:
ORACLE_HOME/opam/jlib/third-party/jersey-bundle-1.18.jar:
ORACLE_HOME/opam/jlib/third-party/jsr311-api.jar:.
/javax.mail.jar EmailNotificationPlugin.java
```

3. Create a jar file:

```
jar -cf EmailNotificationPlugin.jar EmailNotificationPlugin.class
EmailNotificationPlugin.properties
```

After you configure this plug-in for Oracle Privileged Account Manager, the plug-in will send an email to the address that you set-up in custom attributes whenever the configured operation is performed. For example, if you configure this plug-in for the account resource type and checkout operation, then the plug-in will send an email notification whenever a checkout is completed.

Example 18–4 Email Notification Post Plug-In Property File

```
import org.codehaus.jettison.json.JSONException;
import org.codehaus.jettison.json.JSONObject;
import org.codehaus.jettison.json.JSONStringer;
import org.codehaus.jettison.json.JSONArray;

import com.oracle.idm.opam.plugin.interfaces.PostPlugin;
import com.oracle.idm.opam.plugin.context.PluginContext;
import com.oracle.idm.opam.plugin.context.PluginResult;
import java.lang.Thread;
import java.io.*;
import java.util.*;
import javax.mail.*;
import javax.mail.internet.*;
import javax.activation.*;

/* Sample post plugin that sends email notification */
public class EmailNotificationPlugin implements PostPlugin {

    public void runPostPlugin(PluginContext ctx, String reqBodyJSON,
                              String pluginCfgJSON) {

        PluginResult result =
            new PluginResult(reqBodyJSON, PluginResult.CODE_SUCCESS, null);

        try {

            result.appendLog("Starting EmailNotificationPlugin");

            /* Get the resource type and operation name from the context */
```

```
JSONObject opJSON = new JSONObject(ctx.getOperationJSON());
String resourceType = opJSON.getString("resourceType");
String operationName = opJSON.getString("operationName");

/* Get the target name */
JSONObject json = null;
String targetName = null;
if (ctx.getTargetJSON() != null) {
    JSONObject targetJSON = new JSONObject(ctx.getTargetJSON());
    json = targetJSON.getJSONObject("target");
    targetName = json.getString("targetName");
}

/* Get the account name */
String accountName = null;
if (ctx.getAccountJSON() != null) {
    JSONObject accountJSON = new JSONObject(ctx.getAccountJSON());
    json = accountJSON.getJSONObject("account");
    accountName = json.getString("accountName");
}

/* Get which user performed the operation */
JSONObject pluginAuthJSON =
    new JSONObject(ctx.getAuthContextJSON());
String requestor = pluginAuthJSON.getString("requestor");

/* Get custom attributes defined in plugin configuration such as email server,
   to address etc */
JSONObject plugin = new JSONObject(pluginCfgJSON);
JSONObject config = plugin.getJSONObject("plugin");

JSONArray customAttrsArr = new JSONArray();
if (config.has("pluginCustomAttrs"))
    customAttrsArr = config.getJSONArray("pluginCustomAttrs");

String smtpServer = null;
String smtpPort = null;
String fromAddr = null;
String user = null;
String password = null;
JSONArray emailList = null;

for (int i = 0; i < customAttrsArr.length(); i++) {
    JSONObject singleJSON = customAttrsArr.getJSONObject(i);
    JSONObject singleAttr =
        singleJSON.getJSONObject("pluginCustomAttr");

    String attrName = singleAttr.getString("attrname");

    if (attrName.equalsIgnoreCase("smtp_server"))
        smtpServer =
                singleAttr.getJSONArray("attrvalue").getString(0);
    if (attrName.equalsIgnoreCase("smtp_port"))
        smtpPort =
                singleAttr.getJSONArray("attrvalue").getString(0);
    if (attrName.equalsIgnoreCase("from_addr"))
        fromAddr =
                singleAttr.getJSONArray("attrvalue").getString(0);
    if (attrName.equalsIgnoreCase("to_addr"))
        emailList = singleAttr.getJSONArray("attrvalue");
```

```
                if (attrName.equalsIgnoreCase("user"))
                    user = singleAttr.getJSONArray("attrvalue").getString(0);
                if (attrName.equalsIgnoreCase("password"))
                    password =
                            singleAttr.getJSONArray("attrvalue").getString(0);
            }

            for (int i = 0; i < emailList.length(); i++) {
                Properties properties = System.getProperties();
                properties.setProperty("mail.smtps.host", smtpServer);
                properties.setProperty("mail.smtp.port", smtpPort);
                if (user != null && password != null) {
                    properties.setProperty("mail.user", user);
                    properties.setProperty("mail.password", password);
                }
                Session session = Session.getDefaultInstance(properties);

                MimeMessage message = new MimeMessage(session);

                message.setFrom(new InternetAddress(fromAddr));
                message.addRecipient(Message.RecipientType.TO,
                                     new InternetAddress(emailList.getString(i)));

                /* Set the email subject and body */
                String subject =
                    "OPAM Notification : " + resourceType + " " + operationName;
                String emailBody =
                    "Target : " + targetName + "\nAccount : " + accountName +
                    "\nOperation : " + operationName + "\nRequestor : " +
                    requestor;
                message.setSubject(subject);
                message.setText(emailBody);

                /* Send the email */
                Transport.send(message);
                result.appendLog("Completed EmailNotificationPlugin successfully");
                ctx.setPluginResult(result);
            }
        } catch (Exception e) {
            result.appendLog("Exception happened: " + e);

            result.setErrorCode(PluginResult.CODE_FAILURE);
            result.setErrorMsg(e.getMessage());
            ctx.setPluginResult(result);
        }
    }
}
```

Example 18–5 Email Notification Post Plug-In Custom Attributes Property File (EmailNotification.properties)

```
 <customattribute>
    <name value="smtp_server"/>
    <default value=[""]/>
    <mask value="false"/>
    <required value="true"/>
  </customattribute>
 <customattribute>
    <name value="smtp_port"/>
```

```
    <default value=["25"]/>
    <mask value="false"/>
    <required value="true"/>
  </customattribute>
  <customattribute>
    <name value="to_addr"/>
    <default value=[""]/>
    <mask value="false"/>
    <required value="true"/>
  </customattribute>
  <customattribute>
    <name value="from_addr"/>
    <default value=[""]/>
    <mask value="false"/>
    <required value="true"/>
  </customattribute>
  <customattribute>
    <name value="user"/>
    <default value=[""]/>
    <mask value="false"/>
    <required value="false"/>
  </customattribute>
  <customattribute>
    <name value="password"/>
    <default value=[""]/>
    <mask value="true"/>
    <required value="false"/>
  </customattribute>
```

18.6 Managing Plug-Ins

For information about managing plug-ins,

- Refer to Chapter 13, "Working with Plug-Ins" for information about managing plug-ins from the Console.
- Refer to Section A.9, "Working with Plug-Ins" for information about managing plug-ins from the command line.

19

Integrating Oracle Privileged Account Manager with Other Oracle Identity Management Components

This chapter describes how to configure Oracle Privileged Account Manager for integration with commonly used directory and identity management technologies.

> **Note:** For detailed information about integration for the Oracle Identity Management Suite, refer to *Oracle Fusion Middleware Integration Guide for Oracle Identity Management Suite*.

This chapter includes the following sections:

- Section 19.1, "Integrating with Oracle Identity Manager"
- Section 19.2, "Integrating with Oracle Access Management Access Manager"
- Section 19.3, "Integrating with the Credential Store Framework"

19.1 Integrating with Oracle Identity Manager

This section provides information about the Oracle Privileged Account Manager - Oracle Identity Manager integration process.

The topics include:

- Section 19.1.1, "Topology of Oracle Privileged Account Manager Integration with Oracle Identity Manager"
- Section 19.1.2, "Prerequisites for Integration With Oracle Identity Manager"
- Section 19.1.3, "Configuring Oracle Identity Manager for Integration"
- Section 19.1.4, "Running the `opamSetup` Script"
- Section 19.1.5, "Creating the `OPAM_TAGS` and `OPAM_CERT_TAGS` UDF"
- Section 19.1.6, "Tagging Catalog Entries with Oracle Privileged Account Manager Metadata"

Note: If you are using Oracle Privileged Account Manager on IBM WebSphere, refer to "Differences When Integrating with Oracle Identity Manager" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for information about this topic.

19.1.1 Topology of Oracle Privileged Account Manager Integration with Oracle Identity Manager

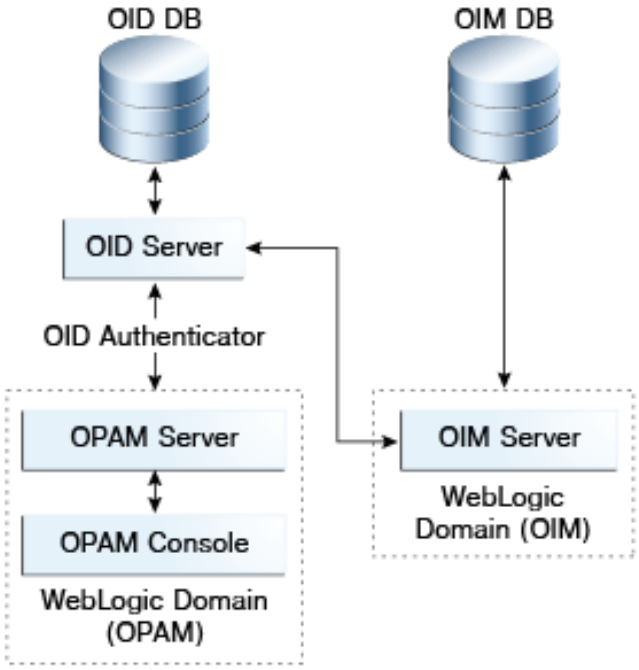
The integration of Oracle Privileged Account Manager and Oracle Identity Manager enables you to manage access to the LDAP groups that are also Oracle Privileged Account Manager grantees. Specifically, integrating these two products enables you to

- Manage the identity lifecycle from hiring to retirement
- Provide a native ability to automate adding and removing users to the proper LDAP groups based on their HR system updates
- Provide the ability to manually request access to accounts
- Support the ability to get approvals for requests
- Support reporting that you can use for attestation reporting; either to augment or in-lieu of Oracle Privileged Account Manager's own reporting.

In addition, Oracle Privileged Account Manager leverages Oracle Identity Manager for workflow support. The integration points include:

- Access to privileged accounts granted to roles in Oracle Privileged Account Manager by an Oracle Privileged Account Manager administrator
- End users can request membership in these roles through Oracle Identity Manager
- Standard Oracle Identity Manager workflow are used to approve these requests
- Membership in the requested role results in end users getting access to the corresponding privileged accounts in Oracle Privileged Account Manager

Figure 19–1 Oracle Identity Manager Workflow Topology



To support this integration, Oracle Identity Manager

- Provides LDAP connector(s) to manage LDAP groups
- Populates the resource catalog with the proper enterprise roles and entitlements.

  Oracle Privileged Account Manager target-accounts are entitlements because Oracle Identity Manager is not actually granting direct access to the actual account only a representation of that account.

> **Note:** If you are using Oracle Privileged Account Manager on IBM WebSphere, refer to "Differences in How Oracle Privileged Account Manager is Deployed in Oracle Fusion Middleware"in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for information.

19.1.2 Prerequisites for Integration With Oracle Identity Manager

This section describes some tasks you must complete before starting the actual integration process. These tasks include:

- Section 19.1.2.1, "Installing Oracle Identity Manager"
- Section 19.1.2.2, "Configuring an Oracle Identity Manager Administrator"
- Section 19.1.2.3, "Configuring the External Identity Stores"
- Section 19.1.2.4, "Creating LDAP Groups"
- Section 19.1.2.5, "Adding the Oracle Privileged Account Manager CA Certificate"

19.1.2.1 Installing Oracle Identity Manager

The instructions in this chapter assume you have already installed Oracle Identity Manager. If you have not yet installed Oracle Identity Manager, refer to the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management* for instructions.

19.1.2.2 Configuring an Oracle Identity Manager Administrator

When you configure an Oracle Identity Manager administrator for this integration, that administrator must be able to perform these tasks:

- Configure an Oracle Identity Manager rule that assigns new users to the proper LDAP groups based on a business rule. The rule should apply whether you assign the new users manually through the user screen or automatically by using an HR/text feed.
- Use Oracle Identity Manager's native functionality to build requests for items in the Oracle Identity Manager resource catalog to ensure that the catalog is properly populated. Oracle Identity Manager enables users to request access to entitlements contained in the Oracle Identity Manager catalog.
- Set approver fields to the proper values. For example, in situations where one employee requests access to the email account of another employee who will be away from the office for an extended period of time.
- Handle "firecall" requests, where an Oracle Privileged Account Manager user must access a system that is outside the normal business process.

  Firecall requests are handled based upon your business requirements and business rules. For example, if the Oracle Privileged Account Manager user is authorized for a target, but the access policy prevents that user from getting the password, then the Oracle Privileged Account Manager administrator can temporarily change the access policy for that target-account.

  If the user cannot wait for Oracle Identity Manager, the Oracle Privileged Account Manager administrator can manually direct access (for example, add a specific grantee to the account) instead.

To review the steps for configuring an Oracle Identity Manager administrator, refer to "Managing Admin Roles" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager*.

19.1.2.3 Configuring the External Identity Stores

You must configure an external identity store as the main authentication source for Oracle Privileged Account Manager. Refer to Section 3.3.2, "Configuring an External Identity Store for Oracle Privileged Account Manager" for more information.

After configuring the Oracle Privileged Account Manager external identity store, you must configure Oracle Identity Manager to use that same identity store.

- You can configure Oracle Identity Manager to use the same LDAP server as Oracle Privileged Account Manager by using LDAPSync.

  Refer to "Completing the Prerequisites for Enabling LDAP Synchronization" in the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management* for instructions.
- Alternatively, you can use the Generic LDAP Connector to configure Oracle Identity Manager to use the same LDAP server as Oracle Privileged Account Manager.

Refer to Section 19.1.3.1, "Installing and Configuring the Generic LDAP Connector" for more information about setting up and configuring the LDAP connector you need for the server.

Note: Best practice is to use LDAPSync for environments where Access Manager is also enabled. If Access Manager is not enabled in your environment, then you can use the LDAP Connector.

In general, most product deployments should use LDAPSync.

19.1.2.4 Creating LDAP Groups

Oracle Privileged Account Manager is optimized for managing shared and privileged accounts, such as `root` on an UNIX system.

Oracle Privileged Account Manager determines which users can check out passwords for accounts on a target, based on the grants those users have received. Grants can be made directly or through membership in *groups*. The groups themselves can be static or dynamic.

Ideally, these LDAP groups should match your enterprise roles. For example, if you have a "Data Center Product UNIX Administrators" enterprise role, you should have a corresponding LDAP group. The benefit of this match is that you can use these groups to control access to other applications besides Oracle Privileged Account Manager target-accounts.

Note: To create an LDAP group, contact your LDAP administrator.

19.1.2.5 Adding the Oracle Privileged Account Manager CA Certificate

You must configure Oracle Privileged Account Manager's Catalog Synchronization task to include the Oracle Privileged Account Manager server's web service Certificate authority (CA) certificate or HTTPS web service calls to the Oracle Privileged Account Manager server cannot succeed.

This process is done in two steps:

1. Retrieve the CA Certificate
2. Import the CA Certificate

Note: If you are using Oracle Privileged Account Manager on an IBM WebSphere server, these steps are slightly different. Refer to "Differences When Integrating with Oracle Identity Manager" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for instructions.

Retrieve the CA Certificate

To retrieve the Oracle Privileged Account Manager server's CA certificate:

1. From your browser, connect to the Oracle Privileged Account Manager server web service:

   https://*hostname*:*sslport*/opam

2. Locate and save the CA certificate (`.pem`) file to the truststore.
   For example, from a Firefox browser

a. Click the lock icon in the browser's address bar. https://

b. When the information dialog box is displayed, click **More information**.

c. On the Page Info dialog, click **View certificate**.

d. On the Certificate Viewer dialog, select the Details tab to view the Certificate Hierarchy.

e. Select the first (root) certificate in the Certificate Hierarchy list, and then click **Export**.

f. When the Save Certificate to File dialog box is displayed, navigate to the directory where you want to save the file. For example, `/tmp/opam.pem`.

g. Select **X.509 Certificate (PEM)** from the Save as type menu, enter **`opam.pem`** as the file name, and click **Save**.

Import the CA Certificate

Run the following command to import the CA certificate file, `opam.pem`, into the WebLogic truststore on the server where you are running Oracle Identity Manager:

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

Where

- *FILE_LOCATION* is the full path and name of the certificate file.
- *ALIAS* with an alias for the certificate.
- *TRUSTSTORE_PASSWORD* is a password for the truststore.
- *TRUSTSTORE_LOCATION* is one of the following truststore paths:

If You Are Using	Then Import the Certificate to the Keystore in this Directory:
Oracle `jrockit_R27.3.1-jdk`	*JROCKIT_HOME*`/jre/lib/security`
Default Oracle WebLogic Server JDK	*WEBLOGIC_HOME*`/java/jre/lib/security/cacerts`
JDK other than Oracle `jrockit_R27.3.1-jdk` or Oracle WebLogic Server JDK	*JAVA_HOME*`/jre/lib/security/cacerts`

19.1.3 Configuring Oracle Identity Manager for Integration

Note: These instructions assume that you have already installed Oracle Identity Manager and that you are an Oracle Identity Manager administrator who can perform the different configuration tasks described in this section.

To prepare Oracle Identity Manager for the integration you must perform the tasks described in the following topics:

- Section 19.1.3.1, "Installing and Configuring the Generic LDAP Connector"
- Section 19.1.3.2, "Creating an Application Instance"

19.1.3.1 Installing and Configuring the Generic LDAP Connector

You must download and install a generic LDAP connector file that works with your LDAP identity store as a target.

For installation instructions, refer to "Installing Connectors" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Manager*.

19.1.3.2 Creating an Application Instance

After installing the connector, you must create an application instance and make it available to Catalog.

For instructions, refer to Part IV, "Application Management," in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Manager*.

19.1.4 Running the `opamSetup` Script

For the Oracle Privileged Account Manager-Oracle Identity Manager integration to become operational, you must run the Oracle Privileged Account Manager-Oracle Identity Manager integration setup script (`opamSetup`), which is available in the following directory:

```
<OIM Oracle Home>/server/bin
```

Run one of the following commands to start the script:

- For **UNIX**, use `opamSetup.sh`
- For **Windows**, use `opamSetup.bat`

```
opamSetup -oimUrl <OIM URL> -oimUser <OIM username>
-oimPassword <OIM user password> -opamItResource <OPAM IT resource name>
-opamServer <OPAM server name> -opamPort <OPAM server port> -opamUser <OPAM user>
-opamPassword <OPAM user password> -idStoreItResource <ID Store IT resource name>
[-ctxFactory <Initial context factory>] [-help]
```

where:

Option	Description
-oimUrl <*OIM URL*>	Provide the URL address for the Oracle Identity Manager server.
-oimUser <*OIM username*>	Provide a Oracle Identity Manager log-in user name.
-oimPassword <*OIM user password*>	Provide the Oracle Identity Manager log-in password.
-opamItResource <*OPAM IT resource name*>	Provide the Oracle Privileged Account Manager IT resource name.
-opamServer <*OPAM server name*>	Provide the path and directory name for the Oracle Privileged Account Manager server.
-opamPort <*OPAM server port*>	Provide the Oracle Privileged Account Manager server port.
-opamUser <*OPAM user*>	Provide a Oracle Privileged Account Manager log-in user name. **Note**: You must be an administrator with the *User Manager* Admin Role and the *Security Administrator* Admin Role to run this command.
-opamPassword <*OPAM user password*>	Provide the Oracle Privileged Account Manager log-in password.
-idStoreItResource <*ID Store IT resource name*>	Provide the name of the IT resource in the identity store.
-ctxFactory <*Initial context factory*>	Provide the name of the context factory (usually `weblogic.jndi.WLInitialContextFactory`).
-help	*Optional*. Display usage options for this command

Note: If you inadvertently omit a parameter, you will be prompted to provide it.

The `opamSetup` script performs the following tasks:

1. Creates the Oracle Privileged Account Manager IT resource with the `opamServer`, `opamPort`, `opamUser`, and `opamPassword` set-up script parameters.
2. Creates an Oracle Privileged Account Manager synchronization scheduled job with the following characteristics:
   - **Name**: Oracle Privileged Account Manager Catalog Synchronization Job. If a job with this name already exists, the job appends `a -1` to the name, then `a -2`, and so on.
   - **Schedule type**: Periodic, runs every 15 minutes.
   - **OPAMServerIdStoreItResource**: The `idStoreItResource` parameter of the set-up script.
   - **OpamServerItResource**: The `opamItResource` parameter of the set-up script.
3. Creates the `OIM.OPAM.Integration` system property (if it does not yet exist) and sets it to `true`.

If any of these tasks fail, the script automatically executes the next task.

19.1.5 Creating the `OPAM_TAGS` and `OPAM_CERT_TAGS` UDF

After setting up the Oracle Privileged Account Manager-Oracle Identity Manager integration environment, you must manually create the `OPAM_TAGS` and `OPAM_CERT_TAGS` user-defined field (*UDF*) in the Oracle Identity Manager catalog. The `OPAM_TAGS` and `OPAM_CERT_TAGS` UDF enable Oracle Privileged Account Manager to search the Oracle Identity Manager catalog.

To manually create the `OPAM_TAGS` and `OPAM_CERT_TAGS` UDF, perform the following steps:

1. Open the Oracle Identity Manager Admin Console and log in to Oracle Identity System Administration.
2. Create and activate a sandbox.

   Note: For detailed instructions about creating and activating a sandbox, refer to the "Managing Sandboxes" section in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.

3. In the left pane, under **System Entities**, click **Catalog** to open the manage Catalog page.
4. Click the **Create a custom field** icon.
5. When the Select Field Type dialog box is displayed, select the **Text** field type to create a text field. Click **OK**.
6. When the page to create a custom field is displayed, specify the following settings:
   - **Appearance section**: Type **`OPAM tags`** in the **Display Label** field.

- **Name section**: Type **`OPAM_TAGS`** in the **Name** field and type **`OPAM metadata tags`** in the **Description** field.
- **Constraints section**: Check the **Searchable** box.
- **Maximum length**: Type **`256`**.
- **Default Value section**: Leave field blank.
- **Advanced section**: Do not check any of the properties boxes.

7. Click **Save and Close**, then verify that the UDFs appear in the custom fields table.
8. Select the Manage Sandboxes tab and click **Publish Sandbox**.

> **Note:** For detailed instructions about publishing a sandbox, refer to the "Managing Sandboxes" section in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.

9. To create the `OPAM_CERT_TAGS` UDF, repeat this procedure with the following changes to Step 6:
   - Replace "OPAM tags" with "OPAM cert tags" as the value of the Display Label field in the Appearance section.
   - Replace `OPAM_TAGS` with `OPAM_CERT_TAGS` as the value of the Name field in the Name section.

19.1.6 Tagging Catalog Entries with Oracle Privileged Account Manager Metadata

The Oracle Privileged Account Manager Catalog Synchronization Job created by the `opamSetup` script tags the catalog entries with the Oracle Privileged Account Manager metadata. This job automatically runs every 15 minutes.

If you need to run the job immediately, instead of waiting for the next cycle to begin, you can manually perform the following steps from the Oracle Identity Manager Admin Console:

1. Click **Scheduler**.
2. When the new screen is displayed, search for and select the **OPAM Catalog Synchronization** job.
3. Click **Run Now**.
4. After the job finishes, click **Refresh**.
5. To verify that the job ran successfully, check the **Job History** view.

> **Note:** If you add new targets or accounts to Oracle Privileged Account Manager, you must run the Oracle Privileged Account Manager Catalog Synchronization Job again.

19.1.6.1 Verifying the Availability of Catalog Entries

The "OPAM Catalog Synchronization Job" queries Oracle Privileged Account Manager for accounts and the associated LDAP groups (entitlements) that have access to those accounts. It uses this information and adds searchable tags (OPAM_TAGS UDF) to the entitlements in the Oracle Identity Manager catalog. In the OIM request user interface,

you can perform catalog searches using the OPAM target type, target name, and account name to find the associated entitlements.

In addition catalog item details page in OIM can be customized to display the tags.

The `OPAM_TAGS` are added in the following format:

```
targettype:targetname:accountname
```

For example:

To search for a "unix" account named "root" on the "prodhost" target, you can search for `*root*` or `*prodhost*` or `unix:prodhost:root` to find the associated entitlements giving access to the account.

19.2 Integrating with Oracle Access Management Access Manager

This section explains how Oracle Access Management Access Manager (Access Manager) integrates with Oracle Privileged Account Manager. Using this integration scenario, you can protect Oracle Privileged Account Manager with Access Manager using a WebGate agent.

The topics in this section include:

- Section 19.2.1, "Prerequisites for Integration With Oracle Access Manager"
- Section 19.2.2, "Enabling Single Sign-On"

19.2.1 Prerequisites for Integration With Oracle Access Manager

Before starting the procedure described in Section 19.2.2, "Enabling Single Sign-On," be aware of the following:

- The instructions assume that you configured Oracle Internet Directory as the identity store; however, other component configurations are possible. Refer to the system requirements and certification documentation on Oracle Technology Network for more information about supported configurations.
- In addition, the instructions describe a specific example of using Access Manager to protect URLs. Although they outline the general approach for this type of configuration, you are not limited to using the exact steps and components described here. For example, Oracle Internet Directory is one of several identity stores certified with Access Manager 11*g*.
- You can use Oracle Adaptive Access Manager as an authentication option with Access Manager. Oracle Adaptive Access Manager provides strong-authentication and risk-based authorization that can be used to provide layered security for Oracle Privileged Account Manager.

  To enable Oracle Adaptive Access Manager with Oracle Privileged Account Manager, select **Access Manager** as the authentication option for the WebGate that is protecting Oracle Privileged Account Manager.
- Oracle Privileged Account Manager is protected by the domain agent out-of-the-box.

19.2.2 Enabling Single Sign-On

By default, the Access Manager 11g agent provides Single Sign On functionality for Oracle Privileged Account Manager and the following Identity Management consoles:

- Oracle Identity Manager

- Access Manager
- Oracle Adaptive Access Manager
- Oracle Authorization Policy Manager

The Access Manager agent can only protect consoles in a single domain. If your environment spans multiple domains, you can use Access Manager 11g WebGate for Oracle HTTP Server 11g.

You can use Access Manager to enable Single Sign On for the Oracle Privileged Account Manager's user interface by using any Access Manager authentication scheme as the challenge method.

The prerequisites are as follows:

- Oracle HTTP Server has been installed.

  When installing the Oracle HTTP Server, deselect **Oracle WebCache** and associated selected components with WebLogic domain (or WebSphere Cell).
- Access Manager 11*g* has been installed and configured properly.
- Oracle HTTP Server 11g has been installed and configured as a front-ending proxy web server for Oracle Privileged Account Manager.
- Access Manager 11g WebGate for Oracle HTTP Server 11g has been installed on the Oracle HTTP Server 11*g*.

  See Also: *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management* for details about installation of the listed components

The steps for enabling Single Sign On in Oracle Privileged Account Manager are as follows:

1. Log in to the Oracle Access Management Console at

   `http://oam_host:port/oamconsole`

2. Click **Application Domains** in the Access Manager section.
3. When the Application Domains page is displayed, click **Search**.
4. In the search results, click **IAM Suite**.
5. Select the Resources tab.
6. Type **`/oinav/**`** in the Resource URL field, and then click **Search**.
7. Verify that the `/oinav/**` URL is listed as a resource in the search results and that its Authentication Policy is listed as a Protected Resource Policy.
   - If the `/oinav/**` URL is not listed, then you must use the Access Management Console to configure a new resource for the agent under which the Oracle Privileged Account Manager URL is to be protected. Refer to Section 19.2.2.1, "Configure a New Resource for the Agent" for more information.
   - If the `/oinav/**` URL is available in the list, then you can proceed to step 8.
8. Configure Oracle HTTP Server to point to the Access Manager domain which has the resources and policies configured. Refer to Section 19.2.2.2, "Configure Oracle HTTP Server for the Access Manager Domain" for more information.
9. Use the Access Management Console to add the following new identity providers:

- Access Manager Identity Asserter
- Oracle Internet Directory Authenticator

Refer to Section 19.2.2.3, "Add New Identity Providers" for more information.

10. Use a WLST command to enable access to more than one application using multiple tabs in a browser session. Refer to Section 19.2.2.4, "Configure Access to Multiple Applications" for more information.

This section also discusses the following topics:

- Section 19.2.2.1, "Configure a New Resource for the Agent"
- Section 19.2.2.2, "Configure Oracle HTTP Server for the Access Manager Domain"
- Section 19.2.2.3, "Add New Identity Providers"
- Section 19.2.2.4, "Configure Access to Multiple Applications"

19.2.2.1 Configure a New Resource for the Agent

After deploying Access Manager, if the `/oinav/**` URL is not available in the Agent IAMSuite's resource list, then you must configure that resource. The high-level steps for this process follow:

> **Note:** For more information, refer to "Defining Resources in an Application Domain" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Access Management*.

1. Log in to the Oracle Access Management Console at

   `http://oam_host:port/oamconsole`

2. Select the **Policy Configuration** tab.
3. Under **Application Domains**, select the agent under which the Oracle Privileged Account Manager URL is to be protected (for example, `-OIMDomain`).
4. Select the Resources tab and click the **add** icon to add a new resource. In the Resources table, enter the Resource Type, Host Identifier, and Resource URL value (`/oinav/…/*`) and click the **Apply** button.
5. Choose Protected Policy or the policy whose authentication schema is the LDAP schema. In the Resources table, click the **add** icon and choose the Oracle Privileged Account Manager URL (`/oinav/…/*`) from the drop-down list.
6. Add the newly defined resource to an Authorization Policy in the Application domain as described in "Defining Authorization Policies for Specific Resources," in the *Oracle Fusion Middleware Administrator's Guide for Oracle Access Management*.

19.2.2.2 Configure Oracle HTTP Server for the Access Manager Domain

Perform these steps to ensure that Oracle HTTP Server front ends the Oracle WebLogic Server container where Oracle Privileged Account Manager is installed.

1. Navigate to the Oracle HTTP Server server `config` directory. For example,

   `/scratch/mydir1/oracle/product/11.1.1/as_1/instances/instance1/config/OHS/ohs1`

2. Locate the `<IfModule mod_weblogic.c>` block in the `mod_wl_ohs.conf` file. Add the host and the port number of the Oracle Privileged Account Manager URL to be protected. For example:

   ```
   MatchExpression /oinav* WebLogicHost=host WebLogicPort=port
   ```

   Where `host` and `port` correspond to the host and port on which the Oracle Privileged Account Manager Console was configured.

3. Restart the Oracle HTTP Server server in the OHS install `bin` directory, for example

   ```
   /scratch/mydir1/oracle/product/11.1.1/as_1/instances/instance1/bin
   ```

   by executing the following command:

   ```
   -./opmnctl restartproc ias=component=ohs1
   ```

19.2.2.3 Add New Identity Providers

Perform these steps to add two new identity providers:

1. Log into the Oracle Access Management Console and navigate to **Security Realms** > **myrealm** > **Providers**.
2. Add the following providers:
   - Access Manager Identity Asserter
   - Oracle Internet Directory Authenticator
3. Set the Control Flag of the Access Manager Identity Asserter to `Required`.
4. Update the following settings in the Oracle Internet Directory Authenticator:
   - Set the Control Flag to `Sufficient`
   - Select the **Provider specific** tab and make the necessary changes, supplying the host, port, and other credentials of the Oracle Internet Directory server. Configure the correct LDAP setting in the Oracle Internet Directory Authenticator.

   The users and Groups in the LDAP will be reflected in the console.
5. Reorder the providers as follows:
   a. Access Manager Identity Asserter
   b. Authenticator
   c. Default Authenticator
   d. Default Identity Asserter
6. Restart Oracle WebLogic Server.
7. Enter the protected Oracle Privileged Account Manager URL, which uses the host and port from the Oracle HTTP Server install:

   ```
   http://OHSHost:OHSPort/oinav/faces/idmNag.jspx
   ```

19.2.2.4 Configure Access to Multiple Applications

When Single Sign On protection is provided by an 11*g* Access Manager Server, you must perform the following steps to configure access to applications using multiple tabs in a single browser session by changing to FORM cache mode.

1. Stop the Access Manager Managed Servers.

2. Execute the following online Access Manager WLST command:

```
configRequestCacheType(type='FORM')
```

3. Restart the Access Manager Managed Servers.

19.3 Integrating with the Credential Store Framework

This section explains how Oracle Privileged Account Manager integrates with Credential Store Framework (CSF).

The topics include:

- Section 19.3.1, "Understanding Oracle Privileged Account Manager-Managed CSF Credentials"
- Section 19.3.2, "Provisioning"
- Section 19.3.3, "Lifecycle Management"
- Section 19.3.4, "Application Consumption"

19.3.1 Understanding Oracle Privileged Account Manager-Managed CSF Credentials

The Credential Store Framework (CSF) is an OPSS component that primarily provides secure storage for credentials. For example, many applications use CSF as a mechanism for storing application credentials.

Oracle Privileged Account Manager enables administrators to identify account credentials to be secured, shared, audited, and managed. In addition, Oracle Privileged Account Manager supports account lifecycle management activities such as periodic password modification.

Though many application developers use CSF to store application credentials for required targets (such as RDBMS and LDAP), there are certain aspects about how CSF is used that can potentially be improved, including:

- Applications storing their credentials in CSF do not expect these credentials to be shared. Therefore, a given instance of CSF can have multiple references to the same credential. For example, multiple applications could be relying on the same physical credential and yet have multiple logical references.
- Periodically modifying application credentials is necessary to satisfy compliance and internal IT policy requirements. However, modifying credentials (on the target and thereafter the CSF reference) remains a manual task, which is further complicated by the fact that there may be multiple references to the same credential in CSF. So, you must change the password or credential on the target and then manually update *all* references to that password in CSF.

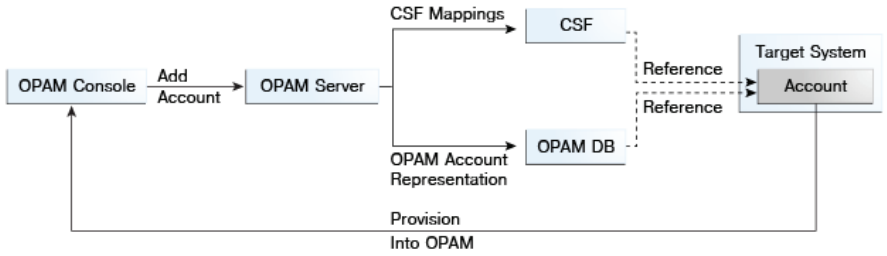
  Oracle Privileged Account Manager can automate this process, but automating the periodic modification of credentials is also complicated by the potential for multiple references that cannot be accurately traced.

Oracle Privileged Account Manager leverages its account lifecycle management feature to empower lifecycle management of application credentials stored in CSF.

19.3.2 Provisioning

If you decide that Oracle Privileged Account Manager will manage a particular account credential, then that credential must be provisioned through Oracle Privileged Account Manager. The following figure illustrates this provisioning process.

Figure 19–2 Oracle Privileged Account Manager Provisioning Process



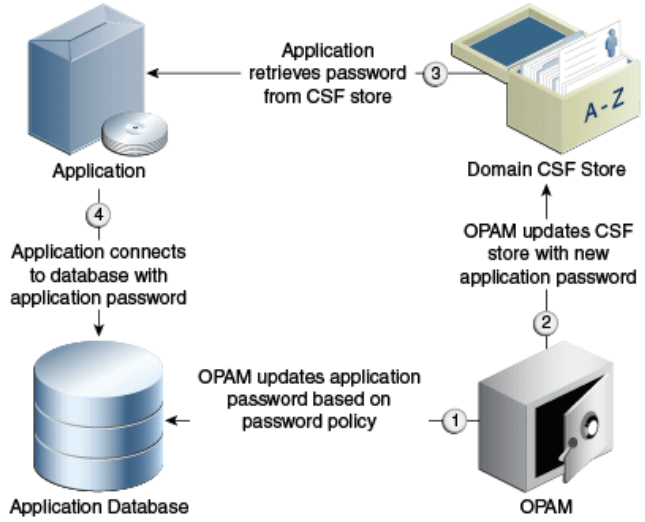
The administrator

1. Adds an Oracle Privileged Account Manager target (if required).
2. Adds the Oracle Privileged Account Manager privileged account or credential to the target, which must include the necessary CSF mappings.

> **Note:** CSF mappings are the mechanism by which a specific credential instance is uniquely identified within CSF.

The Oracle Privileged Account Manager server stores the CSF mappings along with its representation of the privileged account. The Oracle Privileged Account Manager server creates instances of the credential in CSF that correspond to the provided mappings.

Figure 19–3 How Oracle Privileged Account Manager Uses CSF



19.3.3 Lifecycle Management

An account provisioned as described in Section 19.3.2, "Provisioning" can have an associated Password Policy that governs password construction, periodic modification requirements, and so forth.

Oracle Privileged Account Manager normally honors and performs actions on the policy. However, whenever an administrator modifies an account credential that has

associated CSF-mappings, Oracle Privileged Account Manager also updates the credential instances stored in CSF with those mappings (as shown in Figure 19–3). This update ensures that all relevant parties have access to the latest credential and allows the seamless management of password lifecycle events such as periodic modification.

19.3.4 Application Consumption

Using Oracle Privileged Account Manager to manage an application's credentials places no additional burden on that application. The only process change that occurs is that the credential must first be provisioned through Oracle Privileged Account Manager into Oracle Privileged Account Manager and CSF.

Oracle Privileged Account Manager pushes the credential to CSF with the administrator-provided mappings (as shown in Figure 19–3). If those mappings remain constant, the application can continue to access the credentials directly through CSF.

20

Troubleshooting Oracle Privileged Account Manager

This chapter describes common problems that you might encounter when using Oracle Privileged Account Manager and explains how to solve them.

This chapter includes the following sections:

In addition to this chapter, review the *Oracle Fusion Middleware Error Messages Reference* for information about the error messages you may encounter.

20.1 Introduction to Troubleshooting Oracle Privileged Account Manager

This section provides guidelines and a process for using the information in this chapter. Using the following guidelines and process will focus and minimize the time you spend resolving problems.

Guidelines

When using the information in this chapter, Oracle recommends the following:

- After performing any of the solution procedures in this chapter, immediately retry the failed task that led you to this troubleshooting information. If the task still fails when you retry it, perform the procedure of a second solution in this chapter (if provided) and then try the failed task again. Repeat this process until you resolve the problem.
- Make notes about the solution procedures you perform, problems and indications you see, and the data you collect while troubleshooting. If you cannot resolve the problem using the information in this chapter and you must log a service request, the notes you make will expedite the process of solving the problem.

Process

Follow the process outlined in Table 20–1 when using the information in this chapter. If the information in a particular section does not resolve your problem, proceed to the next step in this process.

Table 20–1 Process for Using the Information in this Chapter

Step	Section to Use	Purpose
1	Section 20.2	Get started troubleshooting Oracle Privileged Account Manager. The procedures in this section quickly address a wide variety of problems.
2	Section 20.3	Perform problem-specific troubleshooting procedures for Oracle Privileged Account Manager. This section describes: ▪ Possible causes of the problems ▪ Solution procedures corresponding to each of the possible causes
3	Section 20.5	Use My Oracle Support to get additional troubleshooting information about Oracle Fusion Applications or Oracle BI. My Oracle Support provides access to several useful troubleshooting resources, including Knowledge Base articles and Community Forums and Discussions.
4	Section 20.5	Log a service request if the information in this chapter and My Oracle Support does not resolve your problem. You can log a service request using My Oracle Support at `https://support.oracle.com`.

20.2 Getting Started with Troubleshooting and Logging Basics for Oracle Privileged Account Manager

This section provides information about how to diagnose Oracle Privileged Account Manager problems. The topics include:

- Section 20.2.1, "Increasing the Log Level"
- Section 20.2.2, "Examining Exceptions in the Logs"

20.2.1 Increasing the Log Level

When an Oracle Privileged Account Manager error occurs, you can gather more information about what caused the error by generating complete logs that include debug information and connector logging. the following steps:

1. Set the Oracle Privileged Account Manager logging level to the finest level, which is **`TRACE:32`**.

   Note:

   - For more information about Oracle Privileged Account Manager logging, refer to Chapter 16, "Managing Oracle Privileged Account Manager Auditing and Logging."
   - For more information about setting logging levels, refer to "Implementing Java and Oracle Logging" in the *Oracle Containers for J2EE Developer's Guide*.

2. Repeat the task or procedure where you originally encountered the error.
3. Examine the log information generated using the DEBUG level.

20.2.2 Examining Exceptions in the Logs

Examining the exceptions logged to the Oracle Privileged Account Manager log file can help you identify various problems.

You can access Oracle Privileged Account Manager's diagnostic log in the following directories:

DOMAIN_HOME/servers/Adminserver/logs
DOMAIN_HOME/servers/opamserver/logs

20.3 Resolving Common Problems

This section describes common problems and their solutions. The topics include:

20.3.1 Console Cannot Connect to Oracle Privileged Account Manager Server

Oracle Privileged Account Manager Console cannot connect to the Oracle Privileged Account Manager server.

Cause

If the Console cannot connect to the Oracle Privileged Account Manager server, then you might have a configuration problem with the Console or with Oracle Platform Security Services Trust.

Solution

To resolve this problem:

1. Verify that your host and port information is correct.
2. Confirm that the generated URL displayed on the Console is responsive.
3. Ensure that you correctly completed all of the configuration steps described in "Post-Installation Tasks" in the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*.

Note: If you have configured a high availability instance, ensure that you correctly completed all of the Oracle Privileged Account Manager configuration steps described in the *Oracle Fusion Middleware High Availability Guide*.

20.3.2 Console Changes Are Not Reflected in Other, Open Pages

When you have multiple browser windows or Console tabs open against the same Oracle Privileged Account Manager Console, updates made in one window or tab are not immediately reflected in the other windows or tabs.

Cause

The Oracle Privileged Account Manager Console does not proactively push updates to the browser.

Solution

To resolve this problem, refresh the browser window or tab.

20.3.3 Cannot Access Targets or Accounts

Your attempts to access targets and privileged accounts are failing. You cannot check out, check-in, or test.

Cause 1

The ICF connector being used by Oracle Privileged Account Manager is having issues interacting with the target system.

Solution 1

To resolve this problem:

1. Verify that the target system is up, and that the privileged account of interest exists.
2. Increase Oracle Privileged Account Manager's logging level to **`TRACE:32`** (its finest level) and review the trace logs to determine where the failure occurs.

   Problems are often caused by environmental issues that can be identified using the trace logs and remedied by fixing the configuration on the target system. Refer to Chapter 16, "Managing Oracle Privileged Account Manager Auditing and Logging" for more information.
3. You might have a connector issue. Submit a bug that includes a reproducible test case, target system details, and trace logs.

Cause 2

A user changed the target's service account password out of band from Oracle Privileged Account Manager. For example, if the user changed the password by using the DB host or by using a different Oracle Privileged Account Manager instance in a different domain, the Show Password feature for the original Oracle Privileged Account Manager server does not reflect that change and any attempt to connect to that target will fail.

Solution 2

To resolve this problem, update the new password by editing the target through the Oracle Privileged Account Manager Console or the command line. Refer to Section 9.8, "Managing Privileged Account Passwords" or to Section A.6.8, "`resetpassword` Command" for more information.

20.3.4 Cannot Add Database Targets

This section describes issues that can prevent you from adding database targets:

- Section 20.3.4.1, "Cannot Connect to Oracle Database with `sysdba` Role"
- Section 20.3.4.2, "Cannot Find Special Options for Adding a Database Target"

20.3.4.1 Cannot Connect to Oracle Database with `sysdba` Role

Your attempts to connect to Oracle Database using the `sysdba` role are failing with the following error message:

```
Invalid Connection Details, see server log for details.
```

Cause

To connect to Oracle Database as a user with `sysdba` role, you must configure the **Advanced Properties** option with the value, **`internal_logon=sysdba`**.

You must also specify this setting for the Oracle Database `SYS` account, which must connect with the `sysdba` role. The Oracle Database `SYS` user is a special account and if you do not use this role, then the connection might fail. However, it is a better practice to create a service account instead of using `SYS`.

Solution

To resolve this problem:

1. Connect to Oracle Database as a user with the `sysdba` role.

> **Note:** These configuration steps are not necessary if you are connecting as a normal user.

2. Open the target's General tab and expand **Advanced Configuration** to view the configuration options.
3. Enter the `internal_logon=sysdba` value into the **Connection Properties** field.
4. Click **Test** to retest the connection.
5. **Save** your changes.

20.3.4.2 Cannot Find Special Options for Adding a Database Target

You cannot find configuration options for connecting to database targets such as Oracle RAC Database or for using Secure Socket Layer (SSL).

Cause

Oracle Privileged Account Manager uses a Generic Database connector where special configuration options for specific database target systems are not exposed in a clean or intuitive manner.

Solution

To resolve this problem, define special connectivity options for database targets by modifying the **Database Connection URL** and **Connection Properties** parameter values.

> **Note:**
>
> - Refer to Section 6.2, "Adding and Configuring Targets in Oracle Privileged Account Manager" for information about these parameters.
> - Refer to the *Oracle Identity Manager Connector Guide for Database User Management* for information about which special options are supported.

20.3.5 Cannot Add an Active Directory LDAP Target

An LDAP target using Microsoft Active Directory fails when you test the connection, search for accounts, or check out passwords.

Cause

Active Directory defaults require specific configuration, so you must change the generic default values for the LDAP target. Oracle Privileged Account Manager uses a Generic LDAP connector where special or custom configuration options for specific LDAP target systems are not obvious. Usually, only Active Directory LDAP targets cause issues.

Solution

To resolve this problem, ensure the following when you add an LDAP target:

1. Use SSL to communicate with Active Directory.

- Import the SSL certificates into the WebLogic instance running Oracle Privileged Account Manager. Refer to Section 17.1, "Configuring Oracle Privileged Account Manager to Communicate With Target Systems Over SSL" for more information.
- From the Targets page, set the **TCP Port** to your Active Directory SSL port and enable the **SSL** checkbox.

2. Specify the following "Advanced Configuration" parameters:
   - Set **Password Attribute** to **`unicodepwd`**
   - Set **Advanced Configuration** > **Account Object Classes** to **`top|person|organizationalPerson|user`**.
3. Specify an attribute that is suitable for data in Active Directory, such as `uid` or `samaccountname`, for the **Account User Name Attribute**, **Uid Attribute**, and **LDAP Filter for Retrieving Accounts** configuration parameters.

> **Note:** For more information about setting any of the following parameters, refer to Section 6.2.2.2, "Configuring the LDAP Target Type."

20.3.6 Grantee Cannot Perform a Checkout

A grantee's attempt to checkout an account is failing with an `Insufficient Privileges` error.

Cause

The username is case-sensitive for Oracle Privileged Account Manager grants, but not always for WebLogic authentication.

Solution

To resolve this problem, be sure to enable the **Use Retrieved User Name As Principal** option for the authenticator being used for your production identity store. Refer to Section 3.3.2, "Configuring an External Identity Store for Oracle Privileged Account Manager" for more information.

20.3.7 Cannot View Users or Roles from the Configured Remote Identity Store

When you try to grant to a user or group, you cannot view all users and roles from the configured remote identity store.

Cause 1

The Control flag of the authenticator that corresponds to the identity store containing the user or role is not set to `SUFFICIENT`.

Cause 2

The user or role that you are searching for is not present in the first authenticator listed in the providers list.

Solution

To resolve this problem:

1. Set the Control flag for all necessary authenticators to `SUFFICIENT`.

2. By default, Oracle Privileged Account Manager searches for users and groups in the first authenticator in the Providers list. However, if you set the `virtualize` property in `jps-config.xml` to **`true`**, Oracle Privileged Account Manager fetches the entities from all LDAP authenticators. For example,

```
<serviceInstance name="idstore.ldap" provider="idstore.ldap.provider">
<property name="idstore.config.provider" value="oracle.security.jps.wls
.internal.idstore.WlsLdapIdStoreConfigProvider"/>
<property name="CONNECTION_POOL_CLASS" value=
"oracle.security.idm.providers.stdldap.JNDIPool"/>
<property name="virtualize" value="true"/>
</serviceInstance>
```

   In WebLogic, the `jps-config.xml` file is located in the following location:

```
DOMAIN_HOME/config/fmwconfig
```

20.3.8 Group Membership Changes Are Not Immediately Reflected in Oracle Privileged Account Manager

You have an indirect grant through group membership and updates to that group membership are not immediately reflected in Oracle Privileged Account Manager.

For example, if you assign a user to a Oracle Privileged Account Manager administration role or to a group granted with a Oracle Privileged Account Manager privileged account, you may not be able to view these changes right away.

Cause

WebLogic caches group memberships and identity assertions by default. Therefore, changes in the source location will not be reflected in Oracle Privileged Account Manager until the cache entries are recomputed.

Solution

To resolve this problem, modify the caching settings in your WebLogic Authenticator and Asserter configuration to suit your requirements.

Note: For more information, refer to

- "Optimizing the Group Membership Caches" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*
- "Configuring Identity Assertion Performance in the Server Cache" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*

20.3.9 Cannot Use Larger Key Sizes for Export/Import

You are unable to use key sizes larger than 128-bits for `export` or `import` operations.

Cause

The default JRE installation does not contain the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 6.

Solution

To resolve this problem, apply the JCE patch, available for download from http://www.oracle.com/technetwork/java/javase/downloads/jce-6-download-429243.html

20.3.10 Oracle Privileged Account Manager End Users Gain Privileges They Were Not Explicitly Granted

An Oracle Privileged Account Manager end user can access all of the groups associated with a user, but was not explicitly granted access to those groups.

Cause 1

You granted an Oracle Privileged Account Manager end user access through an LDAP group that uses multiple values as its naming value.

For example, assume you configured an environment that uses CN as its naming attribute and that it contains two groups, A and B. Group A has only one CN value, cn=GroupA and group B has two CN values, cn=GroupA and cn=GroupB.

The Oracle Privileged Account Manager host container (WebLogic or WebSphere) will assert that actual members of GroupA are members of GroupA. However, the host container will also assert that the actual members of GroupB are also members of GroupA, which means that the members of GroupB will inadvertently get the privileges associated with GroupA.

Cause 2

You used nested group memberships.

If group B is a member of group A, and you grant group A access to an Oracle Privileged Account Manager resource, then you implicitly grant this privilege to group B.

Solution

To resolve this problem, you must ensure that group entries in LDAP have only a single value for the naming attribute being used.

20.3.11 Cannot Access MSSQL Server Targets and Accounts

Your attempts to access the MSSQL server database target and accounts are failing. You cannot test, check out, or check-in. Following are two reasons why this problem might occur:

Cause 1

The MSSQL driver `sqljdbc4.jar` is missing.

Cause 2

You might be facing JAVA Bug 7105007, which affects Java Versions: 1.6.0_26 and 1.6.0_29. Refer to http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=7105007.

Solution

To resolve this problem:

1. Ensure MSSQL driver is available for the server as described by the note in Database Type description in Table 6.2.2.1, "Configuring the Database Target".

2. Use JAVA version 1.6.0_30 or higher to avoid encountering the referenced JAVA bug.

20.3.12 Troubleshooting Issues with Using Oracle Database TDE

This section describes issues you might encounter when you are attempting to set-up or to operate Oracle Privileged Account Manager in Oracle Database Transparent Data Encryption (TDE) mode. These issues include:

- Section 20.3.12.1, "TDE Wallet Errors"
- Section 20.3.12.2, "The TDE Wallet is Open, but Columns Are Not Encrypted"

20.3.12.1 TDE Wallet Errors

After enabling TDE mode, you see one of the following error messages:

- `No TDE wallet found`
- `TDE wallet is closed`
- `TDE wallet is undefined`
- `TDE wallet is open but has no master key`
- `Columns are encrypted but TDE wallet is not open`

Cause

The expected TDE wallet status is open.

Solution

To resolve a problem with the TDE wallet, refer to "Enabling Transparent Data Encryption" in the *Oracle Database Advanced Security Administrator's Guide*.

20.3.12.2 The TDE Wallet is Open, but Columns Are Not Encrypted

After setting up TDE, you notice that the TDE wallet is open, but the columns are not encrypted.

Cause

The secure Oracle Privileged Account Manager columns are not encrypted.

Solution

To resolve this problem, perform the steps described in "Configuring Oracle Privileged Account Manager" of the *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*.

For example:

```
sqlplus DEV_OPAM/password1 @IAM_HOME/opam/sql/opamxencrypt.sql
```

20.3.13 Cannot Open Session or Video Recordings

This section describes issues you might encounter when you are attempting to view session recording transcripts or video recordings. These issues include:

- Section 20.3.13.1, "Cannot Access Recordings In the Internet Explorer, Safari, or Firefox 33+ Browsers"
- Section 20.3.13.2, "Cannot Access Recordings in Any Browser"

20.3.13.1 Cannot Access Recordings In the Internet Explorer, Safari, or Firefox 33+ Browsers

You used the Internet Explorer, Safari, or Firefox 33+ browsers to log in to the Oracle Privileged Account Manager Console, but could not view the recording transcript or video recording after following the link from account's checkout history page results

Cause

The Internet Explorer, Safari, or Firefox 33+ browsers mandate key sizes that are greater than 1024 bits, but the out-of-the-box DemoCA and certificates that are generated by Oracle WebLogic Server are 512 bits.

Solution

To workaround this issue, you must generate a self-signed certificate with a key size that is greater than 1024 bits. Use the following steps:

1. Generate a self-signed certificate with a key size of 2048 bits.

   Note: Refer to "Using the Oracle WebLogic Server Java Utilities" in the *Oracle Fusion Middleware Command Reference for Oracle WebLogic Server* for more information.

   ```
   java utils.CertGen -keyfilepass <CAPassword> -certfile <hostname>-cert
   -keyfile <hostname>-key -cn <fully qualified hostname> -strength 2048
   -selfsigned -keyusagecritical false -keyusage digitalSignature,nonRepudiation,
   keyEncipherment,dataEncipherment,keyAgreement,keyCertSign,cRLSign
   ```

   For example:

   ```
   java utils.CertGen -keyfilepass password123 -certfile adc2120745-cert
   -keyfile adc2120745-key -cn adc2120745.example.com -strength 2048
   -selfsigned -keyusagecritical false -keyusage digitalSignature,nonRepudiation,
   keyEncipherment,dataEncipherment,keyAgreement,keyCertSign,cRLSign
   ```

2. Move the key with the `demoidentity` alias to *`demoidentityold`*.

   ```
   cd MW_HOME/wlserver/server/lib

   keytool -list -keystore DemoIdentity.jks
   -storepass DemoIdentityKeyStorePassPhrase

   keytool -changealias -alias demoidentity -destalias demoidentityold
   -keypass DemoIdentityPassPhrase -keystore DemoIdentity.jks
   -storepass DemoIdentityKeyStorePassPhrase

   keytool -list -keystore DemoIdentity.jks
   -storepass DemoIdentityKeyStorePassPhrase
   ```

3. Update the DemoIdentityStore with the certificate and key that you generated in Step 1.

   Note: Refer to "Using the Oracle WebLogic Server Java Utilities" in the *Oracle Fusion Middleware Command Reference for Oracle WebLogic Server* for more information.

   ```
   cd MW_HOME/wlserver/server/lib
   ```

```
java utils.ImportPrivateKey -keystore DemoIdentity.jks
-storepass DemoIdentityKeyStorePassPhrase -keyfile <hostname>-key.pem
-keyfilepass <CAPassword> -certfile <hostname>-cert.pem -alias demoidentity
-keypass DemoIdentityPassPhrase
```

4. Import the certificate that you generated in Step 1 into the `DemoTrust.jks` file.

```
keytool -importcert -v -trustcacerts -file <hostname>-cert.pem
-keystore DemoTrust.jks -storepass DemoTrustKeyStorePassPhrase
-alias <hostname>
```

5. Restart the Oracle WebLogic Server Domain.

> **Note:** For an environment hosted on multiple servers, you must repeat this step for each server. Most importantly, you must copy or duplicate the updates you performed on one server (in `MW_HOME/wlserver/server/lib`) on to the other servers.

20.3.13.2 Cannot Access Recordings in Any Browser

When you try to view a session recording or video recording, the "`This web page is not available`" error message is displayed and you are redirected to a URL that uses "localhost" as the host name.

Cause

The Oracle Privileged Account Manager server URL that was configured under the Oracle Privileged Account Manager Server Configuration has `localhost` defined in the URL. This host name cannot be resolved from external hosts.

Solution

Use the Server Configuration page to change the Oracle Privileged Account Manager server URL to reflect the fully qualified host name for the Oracle Privileged Account Manager server.

20.3.14 Session Checkout Does Not Work, Even After Granting the Account

An end user has been granted access to an account. However, when that user tries to connect as that account through the Oracle Privileged Session Manager the connection is disallowed.

Cause

Although the end user has been granted access to the account, the effective Usage Policy does not include **session** as the **Allowed checkout type**. You must explicitly grant **session** access in the Usage Policy.

Solution

Modify the effective Usage Policy to also grant **session** access.

20.3.15 OPAM Console Login Does Not Work in Internet Explorer 11 Browser

You tried to log into Oracle Privileged Account Manager by using the Console in an Internet Explorer 11 browser. No error messages were reported, however the login was not successful.

Cause

The Oracle Privileged Account Manager login does not work in an Internet Explorer 11 browser.

Workaround

Use a lower version (earlier that release 11) of Internet Explorer or another browser.

Solution

Apply the Oracle Universal Installer (OUI) patch for bug number 18071063 as described in the downloaded patch readme.

To download this patch, login to `https://support.oracle.com`. Select the Patches and Updates tab and search for patch number 18071063.

20.3.16 End User Names Created in Oracle Identity Manager with the "#" Character Cannot Login to Oracle Privileged Account Manager

If you create an end user name in Oracle Identity Manager that contains a pound (#) symbol or character, that user will not be able to log into Oracle Privileged Account Manager.

Cause

WebSphere encodes the pound (#) character in the DN.

Workaround

Avoid using the pound (#) character in end user names that will log into Oracle Privileged Account Manager.

20.3.17 Audit Records Appear in BI Reports After a Long Delay

You notice that there is a long delay before audit records appear in BI Reports.

Cause

Oracle Privileged Account Manager audit records are pushed to the database based on an interval. This interval is specified using the OPSS scripts for auditing.

Solution

You can shorten the interval after which audit records are pushed to the database by using the `setAuditRepository` command provided in the OPSS scripts for auditing. For detailed information about using the `setAuditRepository` command, refer to "OPSS Scripts for Auditing" in the *Oracle Fusion Middleware Application Security Guide*.

In addition, the BI publisher can cache data to improve performance. You can tune or disable the caching settings for the Oracle Privileged Account Manager audit reports in BI Publisher. For detailed information about cache settings, refer to "Setting the Caching Properties" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

20.3.18 The "Failure to Load Windows Connector" Exception Occurs

You notice that the "failure to load windows connector" exception occurs when you start the Oracle Privileged Account Manager server.

Cause

Oracle Privileged Account Manager uses the connector server configuration to search for a connector server on which the Windows connector is successfully deployed. If this connector server is not found, the "failure to load windows connector" exception is displayed.

Solution

Verify if the connector server configuration information for the connector server that hosts the Windows connector is specified correctly. If not, provide the correct connector server configuration information in Oracle Privileged Account Manager and restart the server.

20.3.19 Failure to Add a UNIX Target or Checkout a UNIX Account

A UNIX target fails when you test the connection, search for accounts, or check out passwords. This problem may be caused by one or both of the two following causes:

- **Cause 1**

  The "Sudo Authorization" property is not defined correctly, which is causing errors in the communication with the UNIX system.

  Solution

  The "Sudo Authorization" property needs to defined based on the type of target service account that is used to connect to the UNIX system. You can check if the "Sudo Authorization" property needs to be defined for a target service account, as follows:

  – Check if the account itself has root privileges. For example, check if the account is a root account. If yes, then do not select the "Sudo Authorization" property in the configuration properties while configuring the target service account.

  – Check if the account needs to run sudo authorization to become a root account. For example, if you are using an account named "admin", and you run sudo authorization to change this account to a root account, then select the "Sudo Authorization" property while configuring the target service account.

    However, if you are using a sudo account, then follow the procedure described in section "Creating a Target System SUDO User Account for Connector Operations" of *Oracle Identity Manager Connector Guide for UNIX*, to verify the account.

  – Check if the account has root privileges or if it can run sudo authorization to become a root account. If not, such an account cannot be used as a target service account. Choose an account which has root privileges or can run sudo authorization to become a root account.

- **Cause 2**

  The "Login Shell Prompt" property is not defined correctly, which is causing errors in the communication with the UNIX system.

  Solution

  The Login Shell Prompt defines the prompt that is displayed on the screen while logging into the UNIX system, using the target service account. By default, it is a list of common values as shown in the following example:

  ```
  [$#%>~]
  ```

In the above example, the square brackets form a regular expression to indicate that the prompt could be any one of the listed symbols. Check the following cases to define this property:

- While using a root account, login as the root account and check the prompt.
- When using a sudo account, there could be more one prompt. Login as the sudo account and check the prompt. Then, run sudo authorization to switch to root account, and check the prompt. This prompt may be a different one. Ensure that both values are in the list between square brackets.
- Message of the day could interfere with prompt detection. In some systems, there may be some message printed on the screen when logging into the system.

  For example, you may see the `## This is a production system, use carefully ##` message. This message contains the pound (#) symbol, which may also be present in the Login Shell Prompt configuration. This can cause errors. You must fix the message to remove the characters that are used in Login Shell Prompt configuration.

20.3.20 Copying Password to Clipboard Fails in a HA Environment

The "Copy Password to Clipboard" operation fails in a HA environment.

Cause

The "Copy Password to Clipboard" operation relies on the ZeroClipboard javascript library files, which are not shipped with Oracle Privileged Account Manager. Currently, these library files cannot be located through the load balancer in a HA environment. This is because, in ADF framework, the `<af:resource>` tag "source" attribute does not support the deferred EL expression. Instead, it only accepts the full path URL or a relative URL of the weblogic server.

Solution

Perform the following procedure to workaround this issue:

1. Deploy the ZeroClipboard library files on an instance of Oracle Privileged Account Manager (the computer where Oracle Privileged Account Manager has been installed) as described in Section 17.4.1, "Downloading and Deploying the ZeroClipboard Library Files on the Server."
2. Verify the "ZeroClipboard.js" and the "ZeroClipboard.swf" files using URLs as shown in the following examples:
   - To verify the "ZeroClipboard.js" file run:

     `http://myhost.example.com:2001/ZeroClipboard/ZeroClipboard.js`
   - To verify the "ZeroClipboard.js" file run:

     `http://myhost.example.com:2001/ZeroClipboard/ZeroClipboard.swf`
3. Copy the "oinav.ear" file from the following location to a temporary folder:

   $*ORACLE_HOME*/oinav/modules/oinav.ear_11.1.1.3.0/

   Note: Create a separate copy of this .ear file as backup before you perform the rest of this procedure.

4. Unzip the "oinav.ear" file and the "oiNavApp-war.war" file. Locate the oiNavApp-war folder, within which you must locate the "MyAccount.jsff," "MyChkout.jsff," and "ServerConfig.jsff" files in the following specified locations and make the suggested code changes:

   - Locating and modifying **MyAccount.jsff:**

     In the taskflows/opam/myaccount/ folder, find the "MyAccount.jsff" file and edit the .jsff file in the following manner:

     a. In the 12th line, search for the following text:

     ```
     <af:resourcetype="javascript"source="//ZeroClipboard/ZeroClipboard.js"/
     >
     ```

     b. Replace it with the following text:

     ```
     <af:resource type="javascript"
     source="http://myhost.example.com:2001/ZeroClipboard/ZeroClipboard.js"/
     >
     ```

     Note: In the preceding example, `http://myhost.example.com:2001/ZeroClipboard/ZeroClipboard.js` is an example location of the .js library file. You must replace it with the actual library file location in your environment.

     c. In the 27th line, search for the following text:

     ```
     moviePath : '/ZeroClipboard/ZeroClipboard.swf'
     ```

     d. Replace it with the following text:

     ```
     moviePath : 'http://my
     host.example.com:2001/ZeroClipboard/ZeroClipboard.swf'
     ```

     Note: In the preceding example, `http://myhost.example.com:2001/ZeroClipboard/ZeroClipboard.swf` is an example location of the .swf library file. You must replace it with the actual library file location in your environment.

   - Locating and modifying **MyChkout.jsff:**

     In the taskflows/opam/mychkout/ folder, find the "MyChkout.jsff" file and edit the .jsff file in the following manner:

     a. In the 14th line, search for the following text:

     ```
     source="//ZeroClipboard/ZeroClipboard.js"/>
     ```

     b. Replace it with the following text:

     ```
     source="http://myhost.example.com:2001/ZeroClipboard/ZeroClipboard.js"/
     >
     ```

Note: In the preceding example, `http://myhost.example.com:2001/ZeroClipboard/ZeroClipboard.js` is an example location of the .js library file. You must replace it with the actual library file location in your environment.

c. In the 26th line, search for the following text:

```
moviePath : '/ZeroClipboard/ZeroClipboard.swf'
```

d. Replace it with the following text:

```
moviePath :
'http://myhost.example.com:2001/ZeroClipboard/ZeroClipboard.swf'
```

Note: In the preceding example, `http://myhost.example.com:2001/ZeroClipboard/ZeroClipboard.swf` is an example location of the .swf library file. You must replace it with the actual library file location in your environment.

- Locating and modifying **ServerConfig.jsff:**

  In the taskflows/opam/serverconfig/ folder, find the "ServerConfig.jsff" file and edit the .jsff file in the following manner:

  a. In the 14th line, search for the following text:

  ```
  source="//ZeroClipboard/ZeroClipboard.js"/>
  ```

  b. Replace it with the following text:

  ```
  source="myhost.example.com:2001/ZeroClipboard/ZeroClipboard.js"/>
  ```

  Note: In the preceding example, `http://myhost.example.com:2001/ZeroClipboard/ZeroClipboard.js` is an example location of the .js library file. You must replace it with the actual library file location in your environment.

  c. In the 20th line, search for the following text:

  ```
  moviePath : '/ZeroClipboard/ZeroClipboard.swf'
  ```

  d. Replace it with the following text:

  ```
  moviePath :
  'http://myhost.example.com:2001/ZeroClipboard/ZeroClipboard.swf'
  ```

  Note: In the preceding example, `http://myhost.example.com:2001/ZeroClipboard/ZeroClipboard.swf` is an example location of the .swf library file. You must replace it with the actual library file location in your environment.

5. Recreate the new .war and .ear files to include the changes.

6. Shutdown all weblogic processes and replace the modified "oinav.ear" file in the following location, on all instances of Oracle Privileged Account Manager or on all machines running Oracle Privileged Account Manager:

   $*ORACLE_HOME*/oinav/modules/oinav.ear_11.1.1.3.0/

7. Restart all weblogic process and perform the "update deployment" using the weblogic console.

20.3.21 Error in Loading SAP Classes During the Startup of the Server

The diagnostic log displays a warning saying that SAP classes could not be loaded during server startup.

The following warning is displayed:

```
[ICF][WARN]org.identityconnectors.framework.impl.api.local.LocalConnectorI
nfoManagerImpl:createConnectorInfo() - Unable to load class
org.identityconnectors.sap.SAPConnection$SAPDestinationDataProvider from
bundle file:<path to org.identityconnectors.sap-2.0.0.jar>
```

Cause

The SAP third-party jars are not copied and they are missing while loading the SAP connectors.

Solution

For SAP targets, third-party jars must be copied before loading the SAP connectors. To do so, refer to Section 6.2.4.2, "Copying Third-Party JARs for the SAPUM and SAPUME Targets."

20.3.22 Checkout History Search Results for Pattern Search Do Not Include Recent Session Recordings

You cannot find recent session recordings while searching for a pattern in the Checkout History search.

Cause

Oracle Privileged Account Manager uses Oracle Text Index to index session recordings. The index is synchronized every hour by default, so pattern search may not return the most recent session recordings.

Solution

To include the recent session recordings in pattern search results, you can submit an update index request by calling the following URL:

```
https://<opamhost>:<opamport>/opam/checkout/syncindex
```

> **Note:** Security Administrators, User Managers, and Security Auditors are allowed to update the index.

You can also change the frequency of index update as described in Section 17.5.5, "Managing Oracle Text Index for Session Recordings."

20.3.23 The OPAMAgentService Windows Service Stops

After successful registration of the OPAM Agent, the "OPAMAgentService" service stops instantly.

Cause 1

If the service cannot find the required DLL files on the target system, an exception is thrown and the service is stopped. This error may occur specifically with the Microsoft Windows Server 2008, Microsoft Windows Server 2012, and the Microsoft Windows Server 2012 R2 target systems.

Solution 1

To work around this issue, check the OPAMAgentService log file located at the following relative path:

`\\logs\OpamAgentService_`*`YEAR_MONTH_DAY_HOUR_MINUTE_SECOND`*`.log.`

> **Note:** In the preceding relative path, "YEAR_MONTH_DAY_HOUR_MINUTE_SECOND" is a placeholder and represents the date and time format in which the log is saved.

If this log contains the "Required DLLs could not be found" message, then, refer to Section 8.2.1.1, "Important Notes for Installation on Microsoft Windows Server" to work around this issue.

Cause 2

The Microsoft Windows Operating System version that you are using is not supported.

To work around this error, check the OPAMAgentService log file located at the following relative path:

`\\logs\OpamAgentService_`*`YEAR_MONTH_DAY_HOUR_MINUTE_SECOND`*`.log.`

> **Note:** In the preceding relative path, "YEAR_MONTH_DAY_HOUR_MINUTE_SECOND" is a placeholder and represents the date and time format in which the log is saved.

If this log contains the "ERROR : Uploader Config : No known OS detected ! Exiting Agent" message, then refer to Section 8.2.1, "Reviewing the Supported Components and Important Notes for Installation" to see a list of the supported Microsoft Windows Operating System versions.

Cause 3

The .NET version that you are using is not supported.

To work around this error, check the OPAMAgentService log file located at the following relative path:

`\\logs\OpamAgentService_`*`YEAR_MONTH_DAY_HOUR_MINUTE_SECOND`*`.log.`

> **Note:** In the preceding relative path, "YEAR_MONTH_DAY_HOUR_MINUTE_SECOND" is a placeholder and represents the date and time format in which the log is saved.

If this log contains the "ERROR : Uploader Config : .NET version below 4.0 ! Exiting Agent" message, then ensure that the .NET version is 4.5 by upgrading your existing version or by installing a new instance.

20.3.24 A User is Able to Access the Grants of Another User

An Oracle Privileged Account Manager end user can access the grants associated with another user.

Cause

You have configured multiple authenticators in Weblogic, with control flags set as sufficient and users with same username exist in more than one authenticator.

Solution

You can use one of the following solutions to work around this issue:

- Use only one authenticator and remove the others.
- If you have to use multiple authenticators, remove the duplicate users from the authenticator.

20.3.25 Translation is Missing for Some Attributes in Windows Targets

The localized content or translation is missing for some attributes in Windows targets.

Cause

A connector server configuration has been added to Oracle Privileged Account Manager for the first time and the server has not been restarted.

Translations for connector server properties are picked up during the server start up and cached.

Solution

Restart the server.

20.3.26 Administration Tabs are Missing for Delegated Users

When you are logged in as a delegated user, the administration tabs are missing.

Cause

The username is case-sensitive for Oracle Privileged Account Manager delegations, but may not always be the case for Weblogic authentication.

Solution

To resolve this problem, ensure that you enable the "Use Retrieved User Name As Principal" option for the authenticator that is being used for your production identity store. Refer to Section 3.3.2, "Configuring an External Identity Store for Oracle Privileged Account Manager" for more information about working with an external identity store.

20.4 Frequently Asked Questions

This chapter provides answers to frequently asked questions related to the functionality of Oracle Privileged Account Manager and its features.

1. **How can I test if the ZeroClipboard files are deployed properly on the WebLogic server?**

   Use one of the following methods to check if the ZeroClipboard files are deployed properly on the WebLogic server:

   - Log into the WebLogic Server administration console, click **Deployments,** and check if you can find "ZeroClipboard" in the list of deployments.
   - Open a browser, type the following path, and check if you can view the .js file successfully:

     ```
     http://{YOUR_SERVER_PATH}:{SERVER_PORT}/ZeroClipboard/ZeroClipboard
     .js
     ```

2. **When I configure the password display options, why do I get an error message saying that the files are not loaded properly?**

   This issue can occur if the files are not deployed properly, if the flash plug-in is not installed properly, or if the browser has blocked the flash plug-in.

   To work around this issue, perform the following checks:

   - Log into the WebLogic server administration console and click **Deployments** to see if you can find "ZeroClipboard" in the list of deployments, if not, perform the procedure described in Section 17.4.1, "Downloading and Deploying the ZeroClipboard Library Files on the Server" again.
   - Check if the folder name, Ensure that the folder name is exactly the same as "ZeroClipboard".
   - Use the following link to check if you have installed Flash, if not, you can also install flash using the same link:

     `http://helpx.adobe.com/flash-player.html`
   - Check if your browser has blocked the flash plug-in. Refer to the "Enable Flash Player in your browser" section in the following link to enable the flash plug-in:

     `http://helpx.adobe.com/flash-player.html`
   - After you make any changes, clear the cache of your browser.
   - Log out of Oracle Privileged Account Manager and log in again.

   Sometimes browser updates, flash updates, or bugs in ZeroClipboard can cause problems. You can check the ZeroClipboard Community Forum for more current issue information from the following link, or download the newest ZeroClipboard library files if needed:

   `https://github.com/zeroclipboard/ZeroClipboard/issues`

3. **How can I test if the Flash plug-in is properly installed in my environment?**

   Refer to the "Check if Flash Player is installed on your computer" section in the following link:

   `http://helpx.adobe.com/flash-player.html`

4. **Which browsers support the Copy password to clipboard feature?**

   The ZeroClipboard library v1.*x* works in IE7 or later and most of the other major browsers. It is also fully compatible with Flash Player 10.

20.5 Using My Oracle Support for Additional Troubleshooting Information

You can use My Oracle Support (formerly MetaLink) to help resolve Oracle Fusion Middleware problems. My Oracle Support contains several useful troubleshooting resources, such as:

- Knowledge base articles
- Community forums and discussions
- Patches and upgrades
- Certification information

> **Note:** You can also use My Oracle Support to log a service request.

You can access My Oracle Support at https://support.oracle.com.

Part V

Appendixes and Glossary

This part contains the following appendixes:

- Working with the Command Line Tool
- Working with Oracle Privileged Account Manager's RESTful Interface
- Working with the SSH Connector

A

Working with the Command Line Tool

You can use the Oracle Privileged Account Manager command line tool to perform many of the same tasks you perform by using the Oracle Privileged Account Manager Console. This appendix describes how to launch and work with the Oracle Privileged Account Manager command line tool.

This appendix includes the following sections:

Note:

- You can also use the Oracle Privileged Account Manager RESTful interface to perform many of these tasks. For more information, refer to Appendix B, "Working with Oracle Privileged Account Manager's RESTful Interface."
- The information provided in this appendix is essentially the same whether you are using Oracle Privileged Account Manager on WebLogic or on IBM WebSphere; however, there are a few minor differences.

  Refer to "Differences When Using the Oracle Privileged Account Manager Command Line Tool and REST Interfaces on IBM WebSphere" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management* for more information.
- Globalization support for the Oracle Privileged Account Manager command line tool is not available for this release. The command line tool messages and help are only provided in English.

A.1 Using the Command Line Tool

This section describes how to launch and use the command line tool, and it contains the following sections:

- Section A.1.1, "Launching the Command Line Tool"
- Section A.1.2, "Issuing Commands"

A.1.1 Launching the Command Line Tool

Oracle Privileged Account Manager provides the two following methods for launching the command line tool:

- Section A.1.1.1, "Launching the Command Line Tool from IAM_HOME"
- Section A.1.1.2, "Launching the Command Line Tool from Oracle Privileged Account Manager Client Archive"

In most situations, you can use the instructions in Section A.1.1.1, "Launching the Command Line Tool from IAM_HOME" to launch the command line tool.

However, if you want to use the Oracle Privileged Account Manager command line tool from machines other than the one where you set up Oracle Identity Management middleware, use the instructions in Section A.1.1.2, "Launching the Command Line Tool from Oracle Privileged Account Manager Client Archive."

Note: For security purposes, the Oracle Privileged Account Manager server only responds to SSL traffic.

When you provide the Oracle Privileged Account Manager server target to the Oracle Privileged Account Manager command line tool (or to Oracle Privileged Account Manager's web-based Console), you must provide the SSL endpoint as https://*hostname*:*sslport*/opam.

By default, the WebLogic AdminServer (where the Oracle Privileged Account Manager Console runs) responds to SSL on port 7002 (In IBM WebSphere, the port is 8002). The default Oracle Privileged Account Manager server SSL port is 18102 for both WebLogic and IBM WebSphere. You can use the WebLogic console to check the port for your particular instance.

A.1.1.1 Launching the Command Line Tool from *IAM_HOME*

To launch the Oracle Privileged Account Manager command line tool:

1. Open a command window and set the *ORACLE_HOME* and the *JAVA_HOME* variables to the appropriate path.
   - Set *ORACLE_HOME* to `IAM_HOME`.
   - Set *JAVA_HOME* to the JRE location.
2. Change directory to *`ORACLE_HOME`*`/opam/bin`.
3. At the prompt, type one of the following commands:
   - On **UNIX**, type: `opam.sh`

- On **Windows**, type: `opam.bat`

Invoking the command line tool, automatically connects you to the Oracle Privileged Account Manager server.

You can invoke the Oracle Privileged Account Manager command line tool from a remote client by providing the Oracle Privileged Account Manager server's URL (running on the same machine or on a different machine) in the **`-url`** option.

A.1.1.2 Launching the Command Line Tool from *Oracle Privileged Account Manager Client Archive*

The Oracle Privileged Account Manager client is also available as a standalone `.zip` file, located in the following directory of an Oracle Identity and Access Management suite installation:

IAM_HOME`/opam/tools/opamclient.zip`

Copy the archive and then follow these steps to launch the command line tool:

1. Unzip the archive on the machine where the Oracle Privileged Account Manager client is required.

   Unzipping the `opamclient.zip` file creates a top-level directory named `opamclient`.

2. Set the *OPAMCLIENT_HOME* variable to `<UNZIP_DIR>/opamclient` and set the *JAVA_HOME* variable to the JRE location.

3. At the prompt, type one of the following commands:

   - On **UNIX**, type: `opam.sh`
   - On **Windows**, type: `opam.bat`

   Invoking the command line tool, automatically connects you to the Oracle Privileged Account Manager server.

   You can invoke the Oracle Privileged Account Manager command line tool by providing the Oracle Privileged Account Manager server's URL in the **`-url`** option.

A.1.2 Issuing Commands

Use the following syntax to issue any of the Oracle Privileged Account Manager commands:

Note: When entering commands

- On **UNIX**, type: `opam.sh`
- On **Windows**, type: `opam.bat`

```
[-url <url>] -u <username> [-p <password>] [-debug] -x <opam-command>
```

where:

Option	Description
-url <*url*>	Provide the URL address for the Oracle Privileged Account Manager server. **Note**: If you do not specify a URL for this option, it defaults to `https://hostname:10812/opam`.
-u <*username*>	Provide your log-in user name.
-p <*password*>	Provide your log-in password.
-debug	Enable the debugger log.
-x <*opam-command*>	Run the specified Oracle Privileged Account Manager command.

For example:

```
-url https://hostname:sslport/opam -u <username> [-p <password>] [-debug]
-x checkout -targetname <targetname> -accountname <accountname>
```

Note:

- On a Windows system, you must use double quotes (`"`) instead of single quotes (`'`) for parameters that contain spaces. For example,

  ```
  opam.bat -u sec_admin -p passwd -x showtargetpassword
  -targetname "oracle db"
  ```

- On a UNIX system, you can use single quotes (`'`) for parameters that contain spaces. You can also use special symbols, such as a dollar sign (`$`).

A.2 Working with the Server

The following sections contain information about the commands that you use to manage the Oracle Privileged Account Manager server.

- Section A.2.1, "`getconfig` Command"
- Section A.2.2, "`getserverstatus` Command"
- Section A.2.3, "`modifyconfig` Command"

A.2.1 `getconfig` Command

Use the `getconfig` command to view the OPAM Global Config configuration entry, which enables you to access and manage various Oracle Privileged Account Manager server properties.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x getconfig
```

The following table describes the options you can use with this command:

Option	Description
-configtype <*global/session*>	Specify the configuration type.
[-help]	*Optional*. Displays usage options for this command.

See Also:

`modifyconfig` Command

A.2.2 `getserverstatus` Command

Use the `getserverstatus` command to get the status for an Oracle Privileged Account Manager instance.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x getserverstatus <options>
```

The following table describes the options you can use with this command:

Option	Description
[-help]	*Optional*. Displays usage options for this command.

A.2.3 `modifyconfig` Command

Use the `modifyconfig` command to manage Oracle Privileged Account Manager server properties in the OPAM Global Config configuration entry. You can use this command to perform two types of configuration, *global* and *session*.

Global Configuration Type

The following properties are available for global configuration:

- **policyenforcerinterval:** Interval (in seconds) in which Oracle Privileged Account Manager checks accounts and then automatically checks-in the accounts that have exceeded the expiration time defined in the Usage Policy. (Default is *3600* seconds)
- **passwordcyclerinterval:** Interval (in seconds) in which Oracle Privileged Account Manager checks and then resets the password for any accounts that have exceeded the maximum password age defined in the Password Policy. (Default is *3600* seconds)
- **tdemode:** Flag to request that Oracle Privileged Account Manager use Transparent Data Encryption (TDE) mode or non-TDE mode. For more information, refer to Section 17.2, "Securing Data On Disk."
- **resourceLockWaitTimeout:** The maximum time allowed (in seconds) for an operation to obtain a transaction lock on a resource.
- **targettimeout:** Time (in seconds) allowed to perform the target connectivity test operation.

Session Configuration Type

The following properties are available for the session configuration:

- **updateinterval:** Interval (in seconds) in which the Oracle Privileged Session Manager server checks all of the checked out sessions for expiration and updates their transcripts.
- **opamserverurls:** List of Oracle Privileged Account Manager server URLs to which the Session Manager can connect.
- **maxrecordsize:** Maximum recording size that is allowed per session (in KB). When this quota is reached, the session is automatically terminated.

- **restResponseTimeout:** Maximum time allowed (in seconds) for the OPAM Session manager to complete OPAM Server REST URL invocation.
- **maxauditthreads:** Maximum number of audit threads in the session manager audit pool.
- **maxsessions:** Maximum number of concurrent sessions allowed per session manager server.
- **windowAgentCount:** The number of windows agents that have been deployed.

The following properties are SSH-specific:

- **opamListenPort:** The port on which Session Manager listens for incoming SSH connections.
- **sessioncheckoutinstructions:** The checkout instructions that are presented to users for SSH sessions.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x modifyconfig
<options>
```

The following table describes the options you can use with the `modifyconfig` command:

Option	Description
-configtype <*global*/*session*>	Specify the configuration type.
[-propertyname <*property name*>]	Specify the server property to be modified: ▪ `policyenforcerinterval` ▪ `passwordcyclerinterval` ▪ `tdemode`
[-propertyvalue <*property value*>]	Specify the property value to be modified.
[-help]	*Optional*. Displays usage options for this command.

For example,

```
-x modifyconfig -configtype global -propertyname policyenforcerinterval
  -propertyvalue 600
```

or

```
-x modifyconfig -configtype global -propertyname tdemode
  -propertyvalue true
```

See Also:

`getconfig` Command

A.3 Working with the Connector Server Configuration

The following sections contain information about the commands that you use to manage the Windows connector server:

- Section A.3.1, "addconnectorserverconfig Command"
- Section A.3.2, "deleteconnectorserverconfig Command"

A.3.1 addconnectorserverconfig Command

Use the `addconnectorserverconfig` command to add a new connector server configuration.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x addconnectorserverconfig
<options>
```

The following table describes the options you can use with this command:

Option	Description
-connectorservername <*connector server name*>	Provide a name for the new connector server.
-connectorserverhost <*connector server host*>	Specify the host of the connector server.
-connectorserverport <*connector server port number*>	Specify the server port number.
-connectorserverkey <*connector server key*>	Specify the connector server key.
[-connectorserversslenabled <*true/false*>]	*Optional*. Specify whether to enable ssl for this connector server. ■ **true**: ssl is enabled. ■ **false** (*default*): ssl is disabled.
[-connectorserverdescription <*connector server description*>]	*Optional*. Provide a description for the new connector server.
[-connectorservertimeout <*connector server timeout*>]	Specify the command timeout value in seconds. The timeout value must be a non-positive integer number. The default value is `60`.

For example:

```
-u app_config -p password -x addconnectorserverconfig -connectorservername
connserverconfig_1 -connectorserverhost host.mycompany.com -connectorserverport
7859 -connectorserverkey password
```

A.3.2 deleteconnectorserverconfig Command

Use the `deleteconnectorserverconfig` command to delete an existing connector server configuration.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x deleteconnectorserverconfig
<options>
```

The following table describes the options you can use with this command:

Option	Description
-connectorservername <*connector server name*>	Provide a name for the connector server that is to be deleted.

For example:

```
-u app_config -p password -x deleteconnectorserverconfig -connectorservername
connserverconfig_1
```

A.3.3 testconnectorserverconfig Command

Use the `testconnectorserverconfig` command to test a connector server configuration.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x testconnectorserverconfig
<options>
```

The following table describes the options you can use with this command:

Option	Description
-connectorservername <*connector server name*>	Provide a name for the new connector server.
-connectorserverhost <*connector server host*>	Specify the host of the connector server.
-connectorserverport <*connector server port number*>	Specify the server port number.
-connectorserverkey <*connector server key*>	Specify the connector server key.
[-connectorserversslenabled <*true/false*>]	*Optional*. Specify whether to enable ssl for this connector server. ■ **true**: ssl is enabled. ■ **false** (*default*): ssl is disabled.
[-connectorservertimeout <*connector server timeout*>]	Specify the command timeout value in seconds. The timeout value must be a non-positive integer number. The default value is `60`.

For example:

```
-u app_config -p password1 -x testconnectorserverconfig -connectorservername
connserverconfig_1 -connectorserverhost abc02.example.com -connectorserverport
8759 -connectorserverkey password1 -connectorserversslenabled false
-connectorservertimeout 60
```

A.3.4 retrieveconnectorserverconfig Command

Use the `retrieveconnectorserverconfig` command to retrieve a connector server configuration.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x
retrieveconnectorserverconfig <options>
```

The following table describes the options you can use with this command:

Option	Description
[-connectorservername <*connector server name*>]	Provide a name for the new connector server.
[-connectorserverid <*connector server name*>]	Provide a name for the new connector server.

Note: You can use either the "`-connectorservername`" or "`-connectorserverid`" options to specify the connector server configuration that you want to retrieve.

For example:

```
 -u app_config -p password1 -x retrieveconnectorserverconfig -connectorservername
connserverconfig_1
```

A.3.5 searchconnectorserverconfig Command

Use the `searchconnectorserverconfig` command to search for connector server configuration or configurations.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x searchconnectorserverconfig
<options>
```

The following table describes the options you can use with this command:

Option	Description
-connectorservername <*connector server name*>	Provide a name for the new connector server.
-connectorserverhost <*connector server host*>	Specify the host of the connector server.
[-connectorserverdescription <*connector server description*>]	Provide a description for the new connector server.

For example:

```
-u app_config -p password1 -x searchconnectorserverconfig -connectorservername
connserverconfig
```

A.3.6 modifyconnectorserverconfig Command

Use the `modifyconnectorserverconfig` command to modify a connector server configuration.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x modifyconnectorserverconfig
<options>
```

The following table describes the options you can use with this command:

Option	Description
-connectorservername <*connector server name*>	Provide a name for the new connector server.

Option	Description
-propertyname <*property name*>	Specify the property name of the connector server configuration that is to be modified: connectorservername connectorserverhost connectorserverport connectorserverkey connectorserversslenabled connectorservertimeout
-propertyvalue <*property value*>	Specify the property value of the connector server configuration that is to be modified.
[-force <*true/false*>]	*Optional*. Specify whether to enable ssl for this connector server. ■ **true**: ssl is enabled. ■ **false** (*default*): ssl is disabled.

Note: You must specify all multi-valued attributes in this format: `value1|value2|...`

For example:

```
  -u app_config -p password1 -x modifyconnectorserverconfig -connectorservername
connnserverconfig_1 -propertyname connectorservertimeout -propertyvalue 100 -force
true
```

A.4 Working with Policies

The following sections contain information about the commands that you use when working with Oracle Privileged Account Manager Password Policies and Usage Policies.

A.4.1 `addpasswordpolicy` Command

Use the `addpasswordpolicy` command to add a Password Policy.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x addpasswordpolicy <options>
```

The following table describes the options you can use with this command:

Option	Description
-policyname <*policy name*>	Provide a name for the new Password Policy.
-policystatus <*active/disabled*>	Specify the Password Policy status.
[-description <*policy description*>]	*Optional*. Provide a description of the Password Policy.
[-passwordchangedurationunit <*minutes/hours/days*>]	*Optional*. Specify the password age unit.
[-passwordchangedurationvalue <*password change duration value*>]	*Optional*. Specify the password age value.
[-changeoncheckin <*true/false*>]	*Optional*. Specify whether to change the password when checking in the account using this Password Policy.
[-changeoncheckout <*true/false*>]	*Optional*. Specify whether to change the password when checking out the account using this Password Policy.
[-passwordcharsmin <*password minimum chars number*>]	*Optional*. Specify the minimum character length restriction for the Password Policy.
[-passwordcharsmax <*password maximum chars number*>]	*Optional*. Specify the maximum character length restriction for the Password Policy.
[-passwordalphabeticmin <*password minimum alphabetic chars number*>]	*Optional*. Specify the minimum number of alphabetic characters required for the Password Policy.
[-passwordnumericmin <*password minimum numeric chars number*>]	*Optional*. Specify the minimum number of numeric characters required for the Password Policy.
[-passwordalphanumericmin <*password minimum alphanumeric chars number*>]	*Optional*. Specify the minimum number of alphanumeric characters required for the Password Policy.
[-passworduniquemin <*password minimum unique chars number*>]	*Optional*. Specify the minimum number of unique characters required for the Password Policy.
[-passworduppercasemin <*password minimum uppercase chars number*>]	*Optional*. Specify the minimum number of uppercase characters required for the Password Policy.
[-passwordlowercasemin <*password minimum lowercase chars number*>]	*Optional*. Specify the minimum number of lowercase characters required for the Password Policy.
[-passwordspecialmin <*password minimum special chars number*>]	*Optional*. Specify the minimum number of special characters required for the Password Policy.
[-passwordspecialmax <*password maximum special chars number*>]	*Optional*. Specify the maximum number of special characters allowed for the Password Policy.
[-passwordrepeatedmin <*password minimum repeated chars number*>]	*Optional*. Specify the minimum number of repeated characters allowed for the Password Policy.
[-passwordrepeatedmax <*password maximum repeated chars number*>]	*Optional*. Specify the maximum number of repeated characters allowed for the Password Policy.
[-startingchar <*true/false*>]	*Optional*. Specify whether the first character of the generated password can be a numeric character. If you specify **`true`**, then the password cannot start with a number.
[-isaccountnameallowed <*true/false*>]	*Optional*. Specify whether the generated password can be identical to the account name.
[-requiredchars <*required chars*>]	*Optional*. Specify characters that are required in the generated password. Use the comma (**`,`**) symbol to separate the characters. For example, `a,b,c`.
[-allowedchars <*allowed chars*>]	*Optional*. Specify characters that are allowed in the generated password. Use the comma (**`,`**) symbol to separate the characters. For example, `a,b,c`.

Option	Description
[-disallowedchars <*disallowed chars*>]	*Optional*. Specify characters that are not allowed in the generated password. Use the comma (`,`) symbol to separate the characters. For example, `a,b,c`.
[-help]	*Optional*. Displays usage options for this command.

For example:

```
-url https://hostname:sslport/opam -u opamuser1 -p hr_password123 [-debug]
-x addpasswordpolicy -policyname password_policy_hr -policystatus active
-changeoncheckin true
```

A.4.2 `addusagepolicy` Command

Use the `addusagepolicy` command to add a Usage Policy.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x addusagepolicy <options>
```

The following table describes the options you can use with this command:

Option	Description
-policyname <*policy name*>	Provide a name for the new Usage Policy.
-policystatus <*active/disabled*>	Specify the Usage Policy status.
[-description <*policy description*>]	*Optional*. Provide a description of the Usage Policy.
-dateorduration <*date/duration*>	Set an expiration time based on date or duration.
[-expireddateminutesfromcheckout <*minutes to expiration*>]	*Optional*. Specify the number of minutes until expiration. When a checked-out account with this Usage Policy exceeds the specified duration, Oracle Privileged Account Manager automatically checks-in that account. **Note**: This field becomes a required field if you specify **`duration`** for the `-dateorduration` attribute.
[-expireddate <*expiration date*>]	*Optional*. Specify the expiration date. When an account with this Usage Policy meets this expiration date, Oracle Privileged Account Manager automatically checks-in that account. **Note**: This field becomes a required field if you specify **`date`** for the `-dateorduration` attribute.
Use the following three options to specify at what time the access expires on the expiration date: ■ [-expireddatehour <*expiration hour in expire time*>] ■ [-expireddateminutes <*expiration minutes in expire time*>] ■ [-expireddateamorpm <*am/pm*>]	**Note**: These fields become required fields if you specify **`date`** for the `-dateorduration` attribute. ■ *Optional*. Specify an hour. For example, specify 5 if the expiration time should be 5:00. ■ *Optional*. Specify the minutes. For example, specify 30 if the expiration time should be 5:30. ■ *Optional*. Specify whether the expiration time is a.m. or p.m.
-timezone <*time zone*>]	Specify a time zone for the Usage Policy, including the timezone region. For example, (GMT -6:00) *America/Chicago*.

Option	Description
-usagedates <*dates information of usage policy*>]	Specify the usage dates information for the policy by using the pipe (|) symbol to separate days and the colon (**:**) symbol to separate times. For example, `monday:12:0:am:12:0:am|tuesday:1:15:am:2:35:pm`
-enablerecording <*true/false*>	Set this flag to enable (true) or disable (false) session recording when applying the Usage Policy to a session checkout. (Default is **true**.)
[-help]	*Optional*. Displays usage options for this command.

For example:

```
-url https://hostname:sslport/opam -u opamuser1 -p hr_password123 [-debug]
-x addusagepolicy -policyname usage_policy_fromPMtoAM -policystatus active
-dateorduration duration -expireddateminutesfromcheckout 120
-timezone (GMT -6:00) America/Chicago
monday:12:00:am:12:00:am|tuesday:1:15:am:2:35:pm
```

A.4.3 `modifypasswordpolicy` Command

Use the `modifypasswordpolicy` command to modify a Password Policy.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x modifypasswordpolicy
<options>
```

The following table describes the options you can use with this command:

Option	Description
-policyname <*policy name*>	Specify the Password Policy to be modified.
-propertyname <*property name*>	Specify the property name that you want to modify.
-propertyvalue <*property value*>	Specify the property value that you want to modify.
[-help]	*Optional*. Displays usage options for this command.

For example:

```
-url https://hostname:sslport/opam -u opamuser1 -p hr_password123 [-debug]
-x modifypasswordpolicy -policyname password_policy_hr
-propertyname changeoncheckin -propertyvalue true
```

A.4.4 `modifyusagepolicy` Command

Use the `modifyusagepolicy` command to modify a Usage Policy.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x modifyusagepolicy <options>
```

The following table describes the options you can use with this command:

Option	Description
-policyname <*policy name*>	Specify the Usage Policy to be modified.
-propertyname <*property name*>	Specify the property name that you want to modify.
-propertyvalue <*property value*>	Specify the property value that you want to modify.
-enablerecording <*true/false*>	Set this flag to enable (true) or disable (false) session recording when applying the Usage Policy to a session checkout. (Default is **`true`**.)
[-help]	*Optional*. Displays usage options for this command.

For example:

```
-url https://hostname:sslport/opam -u opamuser1 -p hr_password123 [-debug]
-x modifyusagepolicy -policyname usage_policy_fromPMtoAM
-propertyname changeoncheckin -propertyvalue true
```

A.4.5 `removepasswordpolicy` Command

Use the `removepasswordpolicy` command to remove a Password Policy.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x removepasswordpolicy
<options>
```

The following table describes the options you can use with this command:

Option	Description
-policyname <*policy name*>	Specify the Password Policy to remove.
[-help]	*Optional*. Displays usage options for this command.

For example:

```
-url https://hostname:sslport/opam -u opamuser1 -p hr_password123 [-debug]
-x removepasswordpolicy -policyname password_policy_hr
```

A.4.6 `removeusagepolicy` Command

Use the `removeusagepolicy` command to remove a Usage Policy.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x removeusagepolicy <options>
```

The following table describes the options you can use with this command:

Option	Description
-policyname <*policy name*>	Specify the Usage Policy to remove.
[-help]	*Optional*. Displays usage options for this command.

For example:

```
-url https://hostname:sslport/opam -u opamuser1 -p hr_password123 [-debug]
```

```
-x removeusagepolicy -policyname usage_policy_fromPMtoAM
```

A.4.7 `retrievepasswordpolicy` Command

Use the `retrievepasswordpolicy` command to retrieve a Password Policy.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x retrievepasswordpolicy
<options>
```

The following table describes the options you can use with this command:

Option	Description
-policyname <*policy name*>	Specify the Password Policy to be retrieved.
[-help]	*Optional*. Displays usage options for this command.

For example:

```
-url https://hostname:sslport/opam -u opamuser1 -p hr_password123 [-debug]
-x retrievepasswordpolicy -policyname password_policy_hr
```

A.4.8 `retrieveusagepolicy` Command

Use the `retrievepolicy` command to retrieve a Usage Policy.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x retrieveusagepolicy
<options>
```

The following table describes the options you can use with this command:

Option	Description
-policyname <*policy name*>	Specify the Usage Policy to be retrieved.
[-help]	*Optional*. Displays usage options for this command.

For example:

```
-url https://hostname:sslport/opam -u opamuser1 -p hr_password123 [-debug]
-x retrieveusagepolicy -policyname usage_policy_hr
```

A.5 Working with Targets

The following sections contain information about the commands that you use when working with Oracle Privileged Account Manager targets.

- Section A.5.1, "`addtarget` Command"
- Section A.5.2, "`displayalltargets` Command"
- Section A.5.3, "`modifytarget` Command"
- Section A.5.4, "`removetarget` Command"

A.5.1 addtarget Command

Use the addtarget command to add a target.

Command Syntax:

```
[[-url <url>] -u <username> [-p <password>] [-debug] -x addtarget <options>
```

Oracle Privileged Account Manager supports multiple target types, and each target type has different required and optional parameters. You must specify the target type to see the target-specific options, as follows:

Option	Description
-targettype <ldap | unix | database> <*type-specific attributes*>	Specify the target type to see target-specific attributes.

Note: These options should be discovered at run time, *before* you execute the addtarget command.

The following examples illustrate the commands you can execute to list

Example A–1 Supported Target Types

```
sh opam.sh -url <OPAM url> -u <security admin user>
-p <security admin user password> -x addtarget -help
```

For example, if https://*hostname*:*sslport*/opam is the Oracle Privileged Account Manager server URL, execute the following command:

```
sh opam.sh -url https://hostname:sslport/opam -u sec_admin -p password1
-x addtarget -help
```

Example A–2 Required and Optional Parameters for a Specific Target Type

```
sh opam.sh -url <OPAM url> -u <security admin user>
-p <security admin user password> -x addtarget
-targettype <any supported target type> -help
```

For example, if you are using the LDAP target type with https://*hostname*:*sslport*/opam as the Oracle Privileged Account Manager server URL, execute the following command:

```
sh opam.sh -url https://hostname:sslport/opam -u sec_admin -p password1
-x addtarget -targettype ldap -help
```

Refer to the following sections for a description of the parameters used with the different target types:

- Section A.5.1.1, "ldap Target Type Parameters"
- Section A.5.1.2, "database Target Type Parameters"
- Section A.5.1.3, "unix Target Type Parameters"
- Section A.5.1.4, "lockbox Target Type Parameters"
- Section A.5.1.5, "windows Target Type Parameters"
- Section A.5.1.6, "sapum Target Type Parameters"
- Section A.5.1.7, "sapume Target Type Parameters"

A.5.1.1 ldap Target Type Parameters

The following table describes the ldap target type parameters that you can use with this command.

Option	Description
-targetname <*targetname*>	Provide a name for the target.
-domain <*domain*>	Provide a domain name.
-host <*host*>	Provide the host name.
-port <*port*>	Provide the TCP/IP port number used to communicate with the LDAP server.
[-ssl <*ssl*>]	*Optional*. Specify to connect to the LDAP server using SSL.
-principal <*principal*>	Provide the distinguished name with which to authenticate to the LDAP server.
-credentials <*credentials*>	Provide the principal's password.
[-passwordpolicy] <*password policy name*>	*Optional*. Identify a Password Policy to apply to the target. See the "Note" following this table for more information.
[-passwordpolicyid] <*password policy ID*>	*Optional*. Identify a Password Policy to apply to the target. See the "Note" following this table for more information.
-baseContexts <*baseContexts*> [Multi-Valued]	Specify one or more starting points in the LDAP tree to use when searching the tree. Searches are performed when discovering users from the LDAP server or when looking for groups in which the user is a member.
-accountNameAttribute <*accountNameAttribute*>	Identify the attribute that holds the account's user name.
[-description <*description*>]	*Optional*. Provide a description of the target.
[-organization <*organization*>]	*Optional*. Provide the organization name.
[-uidAttribute <*uidAttribute*>]	*Optional*. Provide the name of the LDAP attribute that is mapped to the `UID` attribute. (Defaults to *`uid`*)
[-accountSearchFilter <*accountSearchFilter*>]	*Optional*. Provide an LDAP filter to control which accounts are returned from the LDAP resource. If you do not specify a filter, then only accounts that include all specified object classes will be returned. (Defaults to *`(uid=*)`*)

Option	Description
[-passwordAttribute <*passwordAttribute*>]	*Optional*. Identify the LDAP attribute that holds the password. When changing a user's password, Oracle Privileged Account Manager sets the new password to this attribute. (Defaults to `userpassword`)
[-accountObjectClasses <*accountObjectClasses*>] [Multi-Valued]	*Optional*. Specify the objectclass or objectclasses to use when creating new user objects in the LDAP tree. When entering more than one objectclass, put each entry on its own line and do not use commas or semicolons to separate multiple object classes. Some objectclasses may require that you specify all objectclasses in the class hierarchy. (Defaults to `"top|person|organizationalPerson|inetOrgPerson"`)
[-force <*true/false*>]	*Optional*. Enable or disable the requirement for connection validation. ■ **true**: Skips connection validation. ■ **false** (*default*): Enforces connection validation.
[-help]	*Optional*. Displays usage options for this command.

Note:

- You can use either `-passwordpolicy <password policy name>` or `-passwordpolicyid <policy ID>` to apply a Password Policy to the target.
- You must specify all multi-valued attributes in this format: `value1|value2|...`

A.5.1.2 database Target Type Parameters

The following table describes the database target type parameters that you can use with this command.

Option	Description
-targetname <*targetname*>	Provide a name for the target.
-domain <*domain*>	Provide a domain name.
-host <*host*>	Provide the host name.
-jdbcUrl <*jdbcUrl*>	Provide the JDBC URL that identifies the target system location. Following are some example URL formats: ■ **For Oracle**: `jdbc:oracle:thin:@<host>:<port>:<sid>` ■ **For MSSQL**: `jdbc:sqlserver://<host>:<port>;database=<database>` ■ **For MySQL**: `jdbc:mysql://<host>:<port>/<database>` ■ **For DB2**: `jdbc:db2://<host>:<port>/<database>` ■ **For Sybase**: `jdbc:sybase:Tds:<host>:<port>/<database>`
-loginUser <*loginUser*>	Provide the Admin User name.
-loginPassword <*loginPassword*>	Provide the Admin User's password.

Option	Description
-dbType <*dbType*>	Specify the database type for which the connector is being used. The connector supports the Oracle, MSSQL, MySQL, DB2, and Sybase database types. **Note**: You can also configure the connector to work against custom database types.
[-description <*description*>]	*Optional*. Provide a description of the target.
[-organization <*organization*>]	*Optional*. Provide the organization name.
[-passwordpolicy] <*password policy name*>	*Optional*. Specify a Password Policy to apply to the target. See the "Note" following this table for more information.
[-passwordpolicyid] <*password policy ID*>	*Optional*. Specify a Password Policy to apply to the target. See the "Note" following this table for more information.
[-passwordrollover] <*passwordrollover*>	*Optional*. Specify whether you want the target's service account password to be rolled over according to the assigned Password Policy. ■ **true**: Rollover the service account password, based on the assigned Password Policy. ■ **false** (*default*): Do not rollover the service account password. **Note**: Password rollover for target service accounts is similar to password expiration for privileged accounts. If a password has not been changed by the expiration date configured in the associated Password Policy, then Oracle Privileged Account Manager will automatically change the password to a randomized value.
[-connectionProperties] <*connectionProperties*>	*Optional*. Specify the connection properties you used when configuring the secured connection. You must use name-value pairs, in the following format: `prop1=val1#prop2=val2..`
[-force <*true/false*>]	*Optional*. Enable or disable the requirement for connection validation. ■ **true**: Skips connection validation. ■ **false** (*default*): Enforces connection validation.
[-help]	*Optional*. Displays usage options for this command.

Note:

- You can use either `-passwordpolicy <password policy name>` or `-passwordpolicyid <policy ID>` to apply a Password Policy to the target.
- You must specify all multi-valued attributes in this format: `value1|value2|...`

A.5.1.3 unix Target Type Parameters

The following table describes the unix target type parameters that you can use with this command.

Option	Description
-targetname <*targetname*>	Provide a name for the target.
-domain <*domain*>	Provide a domain name.

Option	Description
-host <*host*>	Provide the host name.
-loginUser <*loginUser*>	Provide a user name with which to log into the target. For example, **`root`**.
-loginUserpassword <*loginUserpassword*>	Provide a password for the Login user.
-loginShellPrompt <*loginShellPrompt*>	Provide the shell prompt to display when you log into the target. For example, **`$`** or **`#`**.
[-description <*description*>]	*Optional*. Provide a description of the target.
[-organization <*organization*>]	*Optional*. Provide the organization name.
[-passwordpolicy] <*password policy name*>	*Optional*. Specify a Password Policy to apply to the target. See the "Note" following this table for more information.
[-passwordpolicyid] <*password policy ID*>	*Optional*. Specify a Password Policy to apply to the target. See the "Note" following this table for more information.
[-passwordrollover] <*passwordrollover*>	*Optional*. Specify whether you want the target's service account password to be rolled over according to the assigned Password Policy. ▪ **true**: Rollover the service account password, based on the assigned Password Policy. ▪ **false** (*default*): Do not rollover the service account password. **Note**: Password rollover for target service accounts is similar to password expiration for privileged accounts. If a password has not been changed by the expiration date configured in the associated Password Policy, then Oracle Privileged Account Manager will automatically change the password to a randomized value.
[-sudoAuthorization] <*sudoAuthorization*>	*Optional*. Specify whether the user required sudo authorization. ▪ **true**: Do not require sudo authorization. ▪ **false** (*default*): Require sudo authorization for `root` user.
[-commandTimeout <*commandTimeout*>]	*Optional*. Specify the command timeout value in milliseconds. (Defaults to *`120000`*)
[-passwordExpectExpressions <*passwordExpectExpressions*>]	*Optional*. Specify the expressions to be displayed on the target when setting the user's password. For example, if the expressions displayed on running the `passwd` command are, `Enter password:` and `Re-enter password:`, then you can enter the following value for this field: `enter password,re-enter password` **Note**: You can use a regular expression, and the two expressions must be separated by a comma. (Defaults to `new[\s](unix[\s])?password:,new[\s](unix[\s])?password([\s]again)?:`)
[-prePasswdExpectExpression <*prePasswdExpectExpression*>]	*Optional*. Specify the prompt that can be displayed on some targets before the password prompts when running the `passwd` command. You must provide the prompt expression and the expected input value for that expression, separated by a comma. (Defaults to *`None`*)
[-sudopasswordExpectExpressions <*sudoPasswdExpectExpressions*>]	*Optional*. Specify the password prompt to be displayed when running a command in `sudo` mode. (Defaults to *`password:`*)

Option	Description
[-force <*true/false*>]	*Optional*. Enable or disable the requirement for connection validation. ■ **true**: Skips connection validation. ■ **false** (*default*): Enforces connection validation.
[-help]	*Optional*. Displays usage options for this command.

Note:

- You can use either `-passwordpolicy <password policy name>` or `-passwordpolicyid <policy ID>` to apply a Password Policy to the target.
- You must specify all multi-valued attributes in this format: `value1|value2|...`

A.5.1.4 lockbox Target Type Parameters

The following table describes the lockbox target type parameters that you can use with this command.

Option	Description
-targetname <*targetname*>	Provide a name for the target.
-domain <*domain*>	Provide a domain name.
-host <*host*>	Provide the host name.
[-description <*description*>]	*Optional*. Provide a description of the target.
[-organization <*organization*>]	*Optional*. Provide the organization name.
[-help]	*Optional*. Displays usage options for this command.

A.5.1.5 windows Target Type Parameters

The following table describes the windows target type parameters that you can use with this command.

Option	Description
-targetname <*targetname*>	Provide a name for the target.
-host <*host*>	Provide the host name.
-port <*port*>	Provide the TCP or IP port number used to communicate with the LDAP server.
-AdminName <*AdminName*>	Enter the administrator's user name with which the system must authenticate. The setting must be "`username`" if a local account is used and "`domainname\username`" if a domain account is used.
-AdminPassword <*AdminPassword*>	Provide the password that must be used while authenticating the target.
[-description <*description*>]	*Optional.* Provide a description of the target.

Option	Description
[-organization <*organization*>]	*Optional.* Provide the organization name.
[-domain <*domain*>]	*Optional.* Provide a domain name.
[-passwordpolicy] <*password policy name*>	*Optional.* Identify a password policy to apply to the target. See the "Note" following this table for more information.
[-passwordpolicyid] <*password policy id*>	*Optional.* Identify a Password Policy to apply to the target. See the "Note" following this table for more information.
[-connectorserverid <*connectorserverid*>]	*Optional.* Identify a connector server to apply to the target.
[-passwordrollover <*passwordrollover*>]	*Optional.* The default value for this option is `false`.
[-force <*true/false*>]	*Optional*. Enable or disable the requirement for connection validation. ■ **true**: Skips connection validation. ■ **false** (*default*): Enforces connection validation.
[-help]	*Optional*. Displays usage options for this command.

Note:

- You can use either `-passwordpolicy <password policy name>` or `-passwordpolicyid <policy ID>` to apply a Password Policy to the target.
- You must specify all multi-valued attributes in this format: `value1|value2|...`

A.5.1.6 sapum Target Type Parameters

The following table describes the sapum target type parameters that you can use with this command.

Option	Description
-targetname <*targetname*>	Provide a name for the target.
-host <*host*>	Provide the host name.
-user <*user*>	Provide a user name that has permissions to connect to and update passwords of existing accounts in the SAP target.
-password <*password*>	Provide the password of the user account you specified.
-systemNumber <*systemNumber*>	Provide the SAP system number. The default value of this option is `00`

Option	Description
-client <*client*>	Provide the SAP client setting. The default value of this option is `000`
-destination <*destination*>	This value must be unique. This is a mandatory connection parameter that is needed for the SAPJCo to interact with the SAP System. Sample value: `dest123`
-masterSystem <*masterSystem*>	Provide the RFC Destination value that is used for identification of the SAP system. This value must be same as that of the Logical System name. Sample value: `EH6TNLC001` The preceding sample value is based on the following format used in SAP system: <*SYSTEM_ID*>CLNT<*CLIENT_NUM*>
-dummyPassword <*dummyPassword*>	Specify a dummy password. The connector first sets the initial password with the value you specify here and then changes it to the productive password which is sent from the process form.
-enableCUA <*enableCUA*>	Specify `yes` or `no` depending on the following conditions: ■ `yes`: If the target system is SAP CUA. ■ `no`: If the target system is not SAP CUA
-passwordPropagateToChildSystem <*passwordPropagateToChildSystem*>	Specify `yes` or `no` depending on the following conditions: ■ `yes`: If CUA Mode is set to "yes". Specifying the value as yes will allow the connector to propagate user password changes from the SAP CUA parent system to its child systems. ■ `no`: If CUA Mode is set to "no" The default value if this option is `no`, because the default value for CUA Mode is no.
[-description <*description*>]	*Optional.*
[-organization <*organization*>]	*Optional.*
[-domain <*domain*>]	*Optional.*
[-passwordpolicy <*passwordpolicy*>]	*Optional.*
[-connectorserverid <*connectorserverid*>]	*Optional.*
[-passwordrollover <*passwordrollover*>]	*Optional.* The default value for this option is `false`.
[-force <*true/false*>]	*Optional*. Enable or disable the requirement for connection validation. ■ **true**: Skips connection validation. ■ **false** (*default*): Enforces connection validation.

Note: You must specify all multi-valued attributes in this format: `value1|value2|...`

A.5.1.7 sapume Target Type Parameters

The following table describes the sapume target type parameters that you can use with this command.

Option	Description
-targetname <*targetname*>	Provide a name for the target.
-host <*host*>	Provide the host name.
-umeUrl <*umeUrl*>	Provide the SAP UME URL.
-umeUserId <*umeUserId*>	Provide the SAP administrator user id that must be used to connect with the target system.
-umePassword <*umePassword*>	Provide the SAP UME password to connect with target system.
-dummyPassword <*dummyPassword*>	Provide the dummy password.
-logonNameInitialSubstring <*logonNameInitialSubstring*>	Specify `abcdefghijklmnopqrstuvwxyz1234567890` for search operation.
[-description <*description*>]	*Optional.*
[-organization <*organization*>]	*Optional.*
[-domain <*domain*>]	*Optional.*
[-passwordpolicy <*passwordpolicy*>]	*Optional.*
[-connectorserverid <*connectorserverid*>]	*Optional.*
[-passwordrollover <*passwordrollover*>]	*Optional.* The default value for this option is `false`.
[-logSPMLRequest <*logSPMLRequest*>]	*Optional.* Specify `yes` to print the SPML request, otherwise specify `no`. The default value for this option is `no`.
[-force <*true/false*>]	*Optional*. Enable or disable the requirement for connection validation. ■ **true**: Skips connection validation. ■ **false** (*default*): Enforces connection validation.

Note: You must specify all multi-valued attributes in this format: `value1|value2|...`

A.5.2 `displayalltargets` Command

Use the `displayalltargets` command to display a listing of all targets.

Note: You must be an administrator with the *User Manager* Admin Role, the *Security Administrator* Admin Role, or the *Security Auditor* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x displayalltargets <options>
```

The following table describes the options you can use with this command:

Option	Description
[-help]	*Optional*. Displays usage options for this command.

A.5.3 `modifytarget` Command

Use the `modifytarget` command to modify a target.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x modifytarget <options>
```

The following table describes the options you can use with this command:

Option	Description
[-targetid <*targetid*>]	*Optional*. Specify the target GUID value of the target to be modified. **Note**: When you configure a target, Oracle Privileged Account Manager automatically assigns a unique target GUID. Refer to Section 6.2, "Adding and Configuring Targets in Oracle Privileged Account Manager" for more information.
[-targetname <*targetname*>]	*Optional*. Specify the name of the target to be modified.
-propertyname <*propertyname*>	Specify the name of the property that you want to modify.
-propertyvalue <*propertyvalue*>	Specify the property value that you want to modify.
[-force <*true/false*>]	*Optional*. Enables or disables the requirement for connection validation. ■ **true**: Skips connection validation. ■ **false** (*default*): Enforces connection validation.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*targetid*> or <*targetname*> to identify a target. Both values are unique.

A.5.4 `removetarget` Command

Use the `removetarget` command to remove a target.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x removetarget <options>
```

The following table describes the options you can use with this command:

Option	Description
-targetid <*target id*>	Specify the target GUID value of the target to be removed. **Note**: When you configure a target, Oracle Privileged Account Manager automatically assigns a unique target GUID. Refer to Section 6.2, "Adding and Configuring Targets in Oracle Privileged Account Manager" for more information.
[-targetname <*target name*>]	*Optional*. Specify the name of the target to be removed
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*targetid*> or <*targetname*> to identify the target. Both values are unique.

A.5.5 `resettargetpassword` Command

Use the `resettargetpassword` command to manually reset a target service account password. When you execute this command, Oracle Privileged Account Manager returns the target service account details and prompts you to enter a new password.

Note:

- You must be an administrator with the *Security Administrator* Admin Role to execute this command.
- This command is not applicable for the lockbox or ldap target types and will return an "`Operation not supported`" error message.
- Refer to Chapter 7, "Working with Service Accounts" for information about service accounts.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] -x resettargetpassword
```

The following table describes the options you can use with this command:

Option	Description
[-targetid <*target id*>]	*Optional*. Identify the target to be reset.
[-targetname <*target name*>])	*Optional*. Identify the target to be reset.
[-password <*account password*>]	*Optional*. Provide a new password for the target.
[-autogen <*true/false*>]	*Optional*. Use to automatically generate a password, according to account Password Policy. ■ **true**: Enable the system to automatically generate passwords. ■ **false** (*default*): Disable the system's ability to automatically generate passwords. Users must specify passwords.
[-help]	*Optional*. Displays usage options for this command.

Note:

- You use either <*targetid*> or <*targetname*> to identify the target.
- You use either <*password*> or <*autogen*> to create a new password for the target.

A.5.6 `retrievetarget` Command

Use the `retrievetarget` command to get information about a target.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x retrievetarget <options>
```

The following table describes the options you can use with this command:

Option	Description
-targetid <*target id*>	Specify the target GUID value of the target to be retrieved. **Note**: When you configure a target, Oracle Privileged Account Manager automatically assigns a unique target GUID. Refer to Section 6.2, "Adding and Configuring Targets in Oracle Privileged Account Manager" for more information.
[-targetname <*target name*>]	*Optional*. Specify the name of the target to be retrieved.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*targetid*> or <*targetname*> to identify the target. Both values are unique.

A.5.7 `searchtarget` Command

Use the `searchtarget` command to search for a target.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x searchtarget <options>
```

The following table describes the options you can use with this command:

Option	Description
[-targettype <ldap | solaris | oracledb>]	*Optional*. Identify the type of target to search for as LDAP, Solaris, or Oracle DB.
[-domain <*domain*>]	*Optional*. Provide a domain to search.
[-targetname <*target name*>]	*Optional*. Provide the target name to search for.
[-help]	*Optional*. Displays usage options for this command.

A.5.8 `showtargetpassword` Command

Use the `showtargetpassword` command to view the password for a target service account. When you execute this command, Oracle Privileged Account Manager returns the target service account details and the password.

Note:

- You must be an administrator with the *Security Administrator* Admin Role to execute this command.
- This command is not applicable for the lockbox target type and will return an "`Operation not supported`" error message.
- Refer to Chapter 7, "Working with Service Accounts" for information about service accounts.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] -x showtargetpassword
```

The following table describes the options you can use with this command:

Option	Description
[-targetid <*target id*>]	*Optional*. Identify the target for which the password is being reset.
[-targetname <*target name*>])	*Optional*. Identify the name of the target for which the password is being reset.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*targetid*> or <*targetname*> to identify the target.

A.5.9 `showtargetpasswordhistory` Command

Use the `showtargetpasswordhistory` command to view the password history for a target where you have reset the password. When you execute this command, Oracle Privileged Account Manager returns the password history.

Note: You must be an administrator with the *Security Administrator* Admin Role to execute this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] -x showtargetpasswordhistory <options>
```

The following table describes the options you can use with this command:

Option	Description
[-targetid <*target id*>]	*Optional*. Identify the target for which you are searching.
[-targetname <*target name*>])	*Optional*. Identify the name of the target for which you are searching.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*targetid*> or <*targetname*> to identify the target.

A.6 Working with Accounts

The following sections contain information about the commands that you use when working with Oracle Privileged Account Manager privileged accounts.

A.6.1 `addaccount` Command

Use the `addaccount` command to add a privileged account.

> **Note:** You must never use the same account as the service account *and* as a privileged account to be managed by Oracle Privileged Account Manager. Refer to Chapter 7, "Working with Service Accounts" for information about service accounts.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x addaccount <options>
```

The following table describes the options you can use with this command:

Option	Description
[-targetid <*target id*>]	*Optional*. Specify the target GUID value of a configured target. **Note**: When you configure a target, Oracle Privileged Account Manager automatically assigns a unique target GUID. Refer to Section 6.2, "Adding and Configuring Targets in Oracle Privileged Account Manager" for more information.
[-targetname <*target name*>]	*Optional*. Specify the target name of a configured target.
[-password <*account password*>]	*Optional*. Specify a default value for the account password. **Note**: This field becomes a required field if the target type is *lockbox*.
[-description <*account description*>]	*Optional*. Provide a description of the account.
-accountname <*accountname*>	Provide a name for the new account.
[-force <*true/false*>]	*Optional*. Enables or disables the requirement for connection validation. ■ **true**: Skips connection validation. ■ **false** (*default*): Enforces connection validation.
[-help]	*Optional*. Displays usage options for this command.

Note:

- You use either <*targetid*> or <*targetname*> to identify the target. Both values are unique.
- You can use `-password` to set up an account password.

A.6.2 `displayallaccounts` Command

Use the `displayallaccounts` command to display a listing of all accounts.

Note: You must be an administrator with the *User Manager* Admin Role or the *Security Administrator* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x displayallaccounts
<options>
```

The following table describes the options you can use with this command:

Option	Description
[-help]	*Optional*. Displays usage options for this command.

A.6.3 `checkin` Command

Use the `checkin` command to check in privileged accounts.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x checkin <options>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account to be checked-in.
([-accountname <*account name*>] and [-targetname <*target name*>])	*Optional*. Identify the account to be checked-in. **Note**: The (<*accountname*> and <*targetname*>) combination forms a unique pair that can be used to identify a specific account.
[-checkoutid <*checkout ID*>]	Specify the checkout ID.
[-force <*true/false*>]	*Optional*. Enables or disables the ability to force check-in a privileged account. A force check-in enables administrators with the *User Manager* Admin Role to check-in privileged accounts that have been checked-out by other users. ■ **true**: Enables force check-ins. ■ **false**: Disables force check-ins.

Option	Description
[-userid <*userid*>]	*Optional*. Specifies which user is to be force checked-in. Oracle Privileged Account Manager allows multiple users to check out an account at the same time. By providing a userid, the force check-in only applies to the specified user.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*accountid*> or the (<*accountname*> and <*targetname*>) combination to identify the account.

A.6.4 `checkout` Command

Use the `checkout` command to check out privileged accounts.

Note: The `checkout` operation also provides a password for you to use.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x checkout <options>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account to be checked-out.
([-accountname <*account name*>] and [-targetname <*target name*>])	*Optional*. Identify the account to be checked-out. **Note**: The (<*accountname*> and <*targetname*>) combination forms a unique pair that can be used to identify a specific account.
[-checkouttype <*password/session*>]	Specify the type of checkout: ▪ `password` (*default*): Allow users to only check out passwords. ▪ `session`: Allow users to only check out sessions.
[-apiversion <version_numer>]	Specify the API version. Defaults to the latest version.
[-comment <*comment*>]	*Optional*. Provide a comment about the checkout.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*accountid*> or (<*accountname*> and <*targetname*>) to identify the account.

A.6.5 `displaycheckedoutaccounts` Command

Use the `displaycheckedoutaccounts` command to display a listing of a user's checked out accounts.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x displaycheckedoutaccounts
<options>
```

The following table describes the options you can use with this command:

Option	Description
[-help]	*Optional*. Displays usage options for this command.

A.6.6 `modifyaccount` Command

Use the `modifyaccount` command to modify a privileged account.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x modifyaccount <options>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account to be modified.
([-accountname <*account name*>] and [-targetname <*target name*>])	*Optional*. Identify the account to be modified. **Note**: The (<*accountname*> and <*targetname*>) combination forms a unique pair that can be used to identify a specific account.

-propertyname <*propertyname*>	Specify the name of the property that you want to modify.
	Note: To modify an account's Credential Store, you must specify `-propertyname keymap`. Where you must provide the `keymap` property value in the following format:
	`-propertyname keymap [map][key][host:port][user][password]`
	For example,
	`[map][key][t3:\/\/myhost:2001][weblogic][abc123]`

-propertyvalue <*propertyvalue*>	Specify the property value that you want to modify.
[-help]	*Optional*. Displays usage options for this command.

Note:

- To identify an account, you can use either <*accountid*> or (<*accountname*> and <*targetname*>).
- To modify an account's Password Policy, you can use either `-propertyname passwordpolicy -propertyvalue` <*policy name*> or `-propertyname passwordpolicyid -propertyvalue` <*policy ID*>.

A.6.7 `removeaccount` Command

Use the `removeaccount` command to remove a privileged account.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x removeaccount <options>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account to be removed.
([-accountname <*account name*>] and [-targetname <*target name*>])	*Optional*. Identify the account to be removed. **Note**: The (<*accountname*> and <*targetname*>) combination forms a unique pair that can be used to identify a specific account.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*accountid*> or (<*accountname*> and <*targetname*>) to identify the account.

A.6.8 `resetpassword` Command

Use the `resetpassword` command to manually reset the password for an account you have checked out. When you execute this command, Oracle Privileged Account Manager returns the account details and prompts you to enter a new password.

Note: For most users, if the account has already been checked back in, you will get an error.

If you are an administrator with the *Security Administrator* Admin Role, you can use this command to reset a password for both checked out and checked-in accounts.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] -x resetpassword
  [-wallet <wallet files directory>] [-wallet password <wallet password>]
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account to be reset.
([-accountname <*account name*>] and [-targetname <*target name*>])	*Optional*. Identify the account to be reset. **Note**: The (<*accountname*> and <*targetname*>) combination forms a unique pair that can be used to identify a specific account.
[-password <*account password*>]	*Optional*. Provide a new password for the account.
[-autogen <*true/false*>]	*Optional*. Use to automatically generate a password, according to the account Password Policy. ■ **true**: Enable the system to automatically generate passwords. ■ **false** (*default*): Disable the system's ability to automatically generate passwords. Users must specify passwords.
[-help]	*Optional*. Displays usage options for this command.

Note:

- You use either <*accountid*> or (<*accountname*> and <*targetname*>) to identify the account.
- If you use <*accountid*> or (<*accountname*> and <*targetname*>), you must use `-password` or `-autogen`.

A.6.9 `retrieveaccount` Command

Use the `retrieveaccount` command to get information about a privileged account, such as which target the account is on. This information does not include passwords.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x retrieveaccount <options>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account to be retrieved.
([-accountname <*account name*>] and [-targetname <*target name*>])	*Optional*. Identify the account to be retrieved. **Note**: The (<*accountname*> and <*targetname*>) combination forms a unique pair that can be used to identify a specific account.
[-targetname <*target name*>]	*Optional*. Identify the account to be retrieved.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*accountid*> or (<*accountname*> and <*targetname*>) to identify the account.

A.6.10 `searchaccount` Command

Use the `searchaccount` command to search for an account.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x searchaccount <options>
```

The following table describes the options you can use with this command:

Option	Description
[-targettype <ldap | unix | oracledb>]	*Optional*. Identify the account to search for.
[-domain <*account domain*>]	*Optional*. Identify the account to search for.
[-targetname <*target name*>]	*Optional*. Identify the account to search for.
[-help]	*Optional*. Displays usage options for this command.

Note: You can use any combination of `-targettype`, `-domain`, or `-targetname` to identify the account. If you do not provide any of these options, the search returns all accounts.

For example, the following search will return all targets:

```
https://<host name>:<port>/opam/target/search?
```

Whereas, the following search will return all targets whose type contains `ldap` and `org`:

```
https://<host name>:<port>/opam/target/search?type=ldap&org=us
```

A.6.11 `searchcheckouthistory` Command

Use the `searchcheckouthistory` command to search the checkouts for an account that you have checked out previously. When you execute this command, Oracle Privileged Account Manager returns the checkout history.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role or the *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x searchcheckouthistory
<options>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account to search for.
[-accountname <*account name*>]	*Optional*. Identify the account to search for.
[-targetname <*target name*>]	*Optional*. Provide the name of the target.
-fromtime <*from time*>	Specify the time to start searching for checkouts by using one of the following formats: ■ month-day-year-hour-minute-second-timezone ■ UTC in seconds
-totime <*to time*>	Specify the time to stop searching for checkouts by using one of the following formats: ■ month-day-year-hour-minute-second-timezone ■ UTC in seconds
[-uid <*user id*>]	Identify the user to be searched.
[-event <*event*>]	Specify the command executed or a term in the log.
[-size <*size*>]	Specify the number of results to be returned.
[-help]	*Optional*. Displays usage options for this command.

A.6.12 `showpassword` Command

Use the `showpassword` command to view the password for an account that you have checked out. When you execute this command, Oracle Privileged Account Manager returns the account details and the password.

Note: If the account has already been checked back in, you will get an error.

You must be an administrator with the *Security Administrator* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] -x showpassword -accountid <accountid>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account for which the password is being retrieved.
([-accountname <*account name*>] and [-targetname <*target name*>])	*Optional*. Identify the account for which the password is being retrieved. **Note**: The (<*accountname*> and <*targetname*>) combination forms a unique pair that can be used to identify a specific account.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*accountid*> or (<*accountname*> and <*targetname*>) to identify the account.

A.6.13 `showpasswordhistory` Command

Use the `showpasswordhistory` command to view the password history for an account that you have checked out, checked in, or reset the password. When you execute this command, Oracle Privileged Account Manager returns the password history.

Note: You must be an administrator with the *Security Administrator* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] -x showpasswordhistory <options>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account to search for.
[-accountname <*account name*>]	*Optional*. Provide the name of the account to search.
[-targetname <*target name*>]	*Optional*. Provide the name of the target to search.
[-help]	*Optional*. Displays usage options for this command.

A.7 Working with Grantees

The following sections contain information about the commands that you use when working with Oracle Privileged Account Manager grantees.

A.7.1 displayallgroups Command

Use the displayallgroups command to display a listing of all groups.

> **Note:** You must be an administrator with the *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x displayallgroups <options>
```

The following table describes the options you can use with this command:

Option	Description
[-help]	*Optional*. Displays usage options for this command.

A.7.2 displayallusers Command

Use the displayallusers command to display a listing of all users.

> **Note:** You must be an administrator with the *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x displayallusers <options>
```

The following table describes the options you can use with this command:

Option	Description
[-help]	*Optional*. Displays usage options for this command.

A.7.3 `grantgroupaccess` Command

Use the `grantgroupaccess` command to give a group access to a privileged account.

```
[-url <url>] -u <username> [-p <password>] [-debug] -x grantgroupaccess <options>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account to which the group is granted access.
([-accountname <*account name*>] and [-targetname <*target name*>])	*Optional*. Identify the account to which the group is granted access. **Note**: The (<*accountname*> and <*targetname*>) combination forms a unique pair that can be used to identify a specific account.
-groupname <*group name*>	Identify the group to be given access.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*accountid*> or (<*accountname*> and <*targetname*>) to identify the account.

A.7.4 `grantuseraccess` Command

Use the `grantuseraccess` command to give a user access to a privileged account.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x grantuseraccess <options>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account to which the user is granted access.
([-accountname <*account name*>] and [-targetname <*target name*>])	*Optional*. Identify the account to which the user is granted access. **Note**: The (<*accountname*> and <*targetname*>) combination forms a unique pair that can be used to identify a specific account.
-userid <*user id*>	Identify the user to be given access.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*accountid*> or (<*accountname*> and <*targetname*>) to identify the account.

A.7.5 `removegroupaccess` Command

Use the `removegroupaccess` command to remove a group's access to a privileged account.

```
[-url <url>] -u <username> [-p <password>] [-debug] -x removegroupaccess <options>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account where access is being removed
([-accountname <*account name*>] and [-targetname <*target name*>])	*Optional*. Identify the account where access is being removed. **Note**: The (<*accountname*> and <*targetname*>) combination forms a unique pair that can be used to identify a specific account.
-groupname <*group name*>	Identify the group whose access is being removed.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*accountid*> or (<*accountname*> and <*targetname*>) to identify the account.

A.7.6 `removeuseraccess` Command

Use the `removeuseraccess` command to remove a user's access to a privileged account.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x removeuseraccess <options>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify the account where access is being removed.
([-accountname <*account name*>] and [-targetname <*target name*>])	*Optional*. Identify the account where access is being removed. **Note**: The (<*accountname*> and <*targetname*>) combination forms a unique pair that can be used to identify a specific account.
-userid <*user id*>	Identify the user whose access is being removed.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*accountid*> or (<*accountname*> and <*targetname*>) to identify the account.

A.7.7 `retrievegrantees` Command

Use the `retrievegrantees` command to get information about the grantees on a privileged account.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x retrievegrantees <options>
```

The following table describes the options you can use with this command:

Option	Description
[-accountid <*account id*>]	*Optional*. Identify from which account the grantees are to be retrieved.
([-accountname <*account name*>] and [-targetname <*target name*>])	*Optional*. Identify from which account the grantees are to be retrieved. **Note**: The (<*accountname*> and <*targetname*>) combination forms a unique pair that can be used to identify a specific account.
[-help]	*Optional*. Displays usage options for this command.

Note: You use either <*accountid*> or (<*accountname*> and <*targetname*>) to identify the account.

A.7.8 `retrievegroup` Command

Use the `retrievegroup` command to get information about a group.

Note: You must be an administrator with the *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x retrievegroup <options>
```

The following table describes the options you can use with this command:

Option	Description
-groupname <*group name*>	Provide the name of the group to retrieve.
[-help]	*Optional*. Displays usage options for this command.

A.7.9 `retrieveuser` Command

Use the `retrieveuser` command to get information about a user.

Note: You must be an administrator with the *User Manager* Admin Role or the *Security Administrator* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x retrieveuser <options>
```

The following table describes the options you can use with this command:

Option	Description
-userid <*user id*>	Identify the user to be retrieved.
[-help]	*Optional*. Displays usage options for this command.

A.7.10 `searchgroup` Command

Use the `searchgroup` command to search for a group.

> **Note:** You must be an administrator with the *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x searchgroup <options>
```

The following table describes the options you can use with this command:

Option	Description
[-groupname <*group name*>]	*Optional*. Provide the name of the group to search for.
[-description <*description*>]	*Optional*. Provide a description of the group.
[-accountname <*account name*>]	*Optional*. Provide the name of the account to search.
[-targetname <*target name*>]	*Optional*. Provide the name of the target to search.
[-help]	*Optional*. Displays usage options for this command.

A.7.11 `searchuser` Command

Use the `searchuser` command to search for a user.

> **Note:** You must be an administrator with the *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x searchuser <options>
```

The following table describes the options you can use with this command:

Option	Description
[-userid <*user id*>]	*Optional*. Search for the user by the user ID.
[-firstname <*first name*>]	*Optional*. Provide the user's first name.
[-lastname <*last name*>]	*Optional*. Provide the user's last name.
[-accountname <*account name*>]	*Optional*. Provide the name of the account to search.
[-targetname <*target name*>]	*Optional*. Provide the name of the target to search.
[-help]	*Optional*. Displays usage options for this command.

A.8 Working with Resource Group

The following sections contain information about the commands that you use when working with Oracle Privileged Account Manager grantees.

- Section A.8.1, "`addresourcegroup` Command"
- Section A.8.2, "`retrieveresourcegroup` Command"
- Section A.8.3, "`retrieveresourcegroup` Command"

A.8.1 addresourcegroup Command

Use the addresourcegroup command to create a resource group.

Note: You must be an administrator with the *Security Administrator* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x addresourcegroup <options>
```

The following table describes the options you can use with this command:

Option	Description
-resourcegroupname <resource group name>	Specify the resource group name to be created.
[-description < description >]	*Optional.* Specify the resource group description.
[-help]	*Optional.* Displays usage options for this command.

A.8.2 retrieveresourcegroup Command

Use the retrieveresourcegroup command to retrieve the information of a resource group.

Note: You must be an administrator with the *Security Administrator* or *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x retrieveresourcegroup
<options>
```

The following table describes the options you can use with this command:

Option	Description
[-resourcegroupname <resource group name>]	*Optional*. Specify the resource group name to be retrieved.
[-resourcegroupid <resource group id>]	*Optional*. Specify the resource group id to be retrieved.

Option	Description
[-help]	*Optional*. Displays usage options for this command.

A.8.3 `retrieveresourcegroup` Command

Use the `retrieveresourcegroup` command to search the information of a resource group.

> **Note:** You must be an administrator with the *Security Administrator* or *User Manager* Admin Role to successfully run this command.

```
[-url <url>] -u <username> [-p <password>] [-debug] -x searchresourcegroup
<options>
```

The following table describes the options you can use with this command:

Option	Description
[-resourcegroupname <resource group name>]	*Optional*. Specify the resource group name.
[-description < description >]	*Optional*. Specify the resource group description.
[-help]	*Optional*. Displays usage options for this command.

A.8.4 `modifyresourcegroup` Command

Use the `modifyresourcegroup` command to update the information of a resource group.

> **Note:** You must be an administrator with the *Security Administrator* or *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x modifyresourcegroup
<options>
```

The following table describes the options you can use with this command:

Option	Description
[-resourcegroupname <resource group name>]	*Optional*. Specify the resource group name to be updated.
[-resourcegroupid <resource group id>]	*Optional*. Specify the resource group id to be updated.
-propertyname	Specify the property name to be updated
-propertyvalue	Specify the property value.
[-description < description >]	*Optional.* Specify the resource group description.
[-help]	*Optional*. Displays usage options for this command.

A.8.5 `removeresourcegroup` Command

Use the `removeresourcegroup` to delete a resource group.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x removeresourcegroup
<options>
```

The following table describes the options you can use with this command:

Option	Description
[-resourcegroupname <resource group name>]	*Optional*. Specify the resource group name to be retrieved.
[-resourcegroupid <resource group id>]	*Optional*. Specify the resource group id to be retrieved.
[-help]	*Optional*. Displays usage options for this command.

A.8.6 `addresourcegroupmember` Command

Use the `addresourcegroupmember` command to add a member into the resource group.

> **Note:** You must be an administrator with the *Security Administrator* or *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x addresourcegroupmember
<options>
```

The following table describes the options you can use with this command:

Option	Description
[-resourcegroupname <resource group name>]	*Optional*. Specify the resource group name.
[-resourcegroupid <resource group id>]	*Optional*. Specify the resource group id.
[-memberid <member id>]	*Optional.* Specify the member id to be added into resource group.
[membertype <member type>]	*Optional.* Specify the member type.
[targetname <target name>]	*Optional.* Specify the target name to be added into resource group.
[accountname <account name>]	*Optional.* Specify the account name to be added into resource group.
[childresourcegroupname < child resource group name>]	*Optional.* Specify the child resource group name to be added into resource group.
[-help]	*Optional*. Displays usage options for this command.

Pass either (<memberid> and <memberType>), (<accountname> and <targetname>), <targetname> or <childresourcegroupname> to identify the member to add.

A.8.7 removeresourcegroupmember Command

Use the removeresourcegroupmember command to delete a member from the resource group.

> **Note:** You must be an administrator with the *Security Administrator* or *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x removeresourcegroupmember
<options>
```

The following table describes the options you can use with this command:

Option	Description
[-resourcegroupname <resource group name>]	*Optional*. Specify the resource group name.
[-resourcegroupid <resource group id>]	*Optional*. Specify the resource group id.
[-memberid <member id>]	*Optional*. Specify the member id to be added into resource group.
[targetname <target name>]	*Optional*. Specify the target name to be added into resource group.
[accountname <account name>]	*Optional*. Specify the account name to be added into resource group.
[childresourcegroupname < child resource group name>]	*Optional*. Specify the child resource group name to be added into resource group.
[-help]	*Optional*. Displays usage options for this command.

Pass either <memberid> or (<accountname> and <targetname>) or <targetname> or <childresourcegroupname> to identify the member to be deleted.

A.8.8 adddelegation Command

Use the adddelegation command to add delegations to the resource group.

> **Note:** You must be an administrator with the *Security Administrator* or *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x adddelegation <options>
```

The following table describes the options you can use with this command:

Option	Description
[-resourcegroupname <resource group name>]	*Optional*. Specify the resource group name.
[-resourcegroupid <resource group id>]	*Optional*. Specify the resource group id.
-delegatee <delegatee>	Specify the delegatee to be added to resource group.
delegateetype <user/group>	Specify the delegatee type.
delegateepriv <delegatee privilege>	Specify the delegatee privilege.
[-help]	*Optional*. Displays usage options for this command.

A.8.9 `removedelegation` Command

Use the `removedelegation` command to delete delegations from the resource group.

> **Note:** You must be an administrator with the *Security Administrator* or *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x removedelegation <options>
```

The following table describes the options you can use with this command:

Option	Description
[-resourcegroupname <resource group name>]	*Optional*. Specify the resource group name.
[-resourcegroupid <resource group id>]	*Optional*. Specify the resource group id.
-delegatee <delegatee>	Specify the delegatee to be added to resource group.
delegateetype <user/group>	Specify the delegatee type.
delegateepriv <delegatee privilege>	Specify the delegatee privilege.
[-help]	*Optional*. Displays usage options for this command.

A.8.10 `retrievedelegation` Command

Use the `deleteresourcegroupmember` command to delete delegations from the resource group.

> **Note:** You must be an administrator with the *Security Administrator* or *User Manager* Admin Role to successfully run this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x retrievedelegation
<options>
```

The following table describes the options you can use with this command:

Option	Description
[-resourcegroupname <resource group name>]	*Optional*. Specify the resource group name.
[-resourcegroupid <resource group id>]	*Optional*. Specify the resource group id.
-delegatee <delegatee>	Specify the delegatee to be added to resource group.
delegateetype <user/group>	Specify the delegatee type.
delegateepriv <delegatee privilege>	Specify the delegatee privilege.
[-help]	*Optional*. Displays usage options for this command.

A.9 Working with Plug-Ins

The following sections describe the commands that you can use to configure and deploy Java plug-ins for Oracle Privileged Account Manager.

- Section A.9.1, "`addplugin` Command"
- Section A.9.2, "`addplugincustomattr` Command"
- Section A.9.3, "`removeplugincustomattr` Command"
- Section A.9.5, "`retrieveplugincustomattr` Command"
- Section A.9.6, "`searchplugin` Command"
- Section A.9.7, "`modifyplugin` Command"
- Section A.9.8, "`removeplugin` Command"

A.9.1 `addplugin` Command

Use the `addplugin` command to add a plug-in to a resource.

> **Note:** You must be an administrator with the *Application Configurator* Admin Role to execute this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x addplugin
```

The following table describes the options you can use with this command:

> **Note:** Oracle Privileged Account Manager uses some of these options as filtering rules to decide whether to execute the plug-in. In addition, Oracle Privileged Account Manager evaluates these filtering rules in a certain order to decide one rule's precedence over another.
>
> For more information, about the filtering rules and creating plug-in configurations, refer to Section 13.2.8, "Filtering Rules." and Section 13.3, "Creating a Plug-In Configuration" respectively.

Option	Description
-pluginname <*plugin name*>	Specify a name for the new plug-in.
-resource <*target/account/server*>	Identify the resource on which the plug-in will perform.
-operation <*plugin operation*>	Specify the operation the plug-in will perform. **Note**: Refer to Section 13.2.7, "Supported Operations and Timings" for a complete list of supported operations.
-timing <*pre/post*>	Specify the plug-in timing. ▪ **Pre-plug-in**: Performed before the Oracle Privileged Account Manager operation. ▪ **Post-plug-in**: Performed after the Oracle Privileged Account Manager operation.
-order <*plugin order*>	Specify the order in which the plug-in should be queued for execution. Where the smaller the number, the closer to the top (or beginning) of the queue. (Minimum value is 1.)
-classname <*plugin class name*>	Specify the plug-in's class name.
-classpath <*plugin class path*> [Multi-Valued]	Specify the path to the plug-in's jar file.
[-description] <*plugin description*>	*Optional*. Provide a description of the plug-in.
[-status] <*active/disabled*>	Specify the plug-in execution status. Where ▪ **active**: Allows the plug-in to execute at runtime. ▪ **disabled**: Does not allow the plug-in to execute at runtime.
[-enableuser] <*plugin enabled user*> [Multi-Valued]	*Optional*. Add one or more users to the plug-in's enabled user list. If the logged in user belongs to the enabled user list, then Oracle Privileged Account Manager will execute the plug-in.
[-disableuser] <*plugin disabled user*> [Multi-Valued]	*Optional*. Add one or more users to the plug-in's disabled user list. If the logged in user belongs to the disabled user list, then Oracle Privileged Account Manager will not execute the plug-in.
[-enablegroup] <*plugin enabled group*> [Multi-Valued]	*Optional*. Add one or more groups to the plug-in's enabled group membership list. If the logged in user belongs to the enabled user membership group, then Oracle Privileged Account Manager will execute the plug-in.
[-disablegroup] <*plugin disabled group*> [Multi-Valued]	*Optional*. Add one or more groups to the plug-in's disabled group membership list. If the logged in user belongs to a disabled membership group, then Oracle Privileged Account Manager will not execute the plug-in.
[-enableresourcegroup] <*plugin enabled resource group*> [Multi-Valued]	*Optional*. Add one or more resource groups to the plug-in's enabled resource group membership list. If the resource belongs to the enabled resource group, then Oracle Privileged Account Manager will execute the plug-in.
[-disableresourcegroup] <*plugin disabled resource group*> [Multi-Valued]	*Optional*. Add one or more resource groups to the plug-in's disabled resource group membership list. If the resource belongs to the disabled resource group, then Oracle Privileged Account Manager will not execute the plug-in.

Option	Description
[-enablehttpresult] <*plugin enabled HTTP result*> [Multi-Valued]	*Optional*. Specify the enabled HTTP response.
[-disablehttpresult] <*plugin disabled HTTP result*> [Multi-Valued]	*Optional*. Specify the disabled HTTP response.
[-version] <*plugin version*>	*Optional*. Specify the plug-in version.
[-timeout] <*plugin timeout*>	*Optional*. Specify the plug-in timeout.
[-help]	*Optional*. Displays usage options for this command.

Note: You must specify all multi-valued attributes in this format: `value1|value2|...`

A.9.2 `addplugincustomattr` Command

Use the `addplugincustomattr` command to add a plug-in custom attribute.

Note: You must be an administrator with the *Security Administrator* Admin Role or the *Application Configurator* Admin Role to execute this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug]  -x addplugincustomattr
```

The following table describes the options you can use with this command:

Option	Description
-pluginname <*plugin name*>	Identify the plug-in on which to add the custom attribute.
-pluginattrname <*plugin custom attribute name*>	Specify the name of the custom attribute.
-pluginattrvalue <*plugin custom attribute value*> [Multi-Valued]	Specify the value of the custom attribute.
[-help]	*Optional*. Displays usage options for this command.

Note: You must specify all multi-valued attributes in this format: `value1|value2|...`

A.9.3 `removeplugincustomattr` Command

Use the `removeplugincustomattr` command to remove a custom attribute from a plug-in.

Note: You must be an administrator with the Security Administrator Admin Role or the *Application Configurator* Admin Role to execute this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug]  -x removeplugincustomattr
```

The following table describes the options you can use with this command:

Option	Description
-pluginname <*plugin name*>	Identify the plug-in from which the custom attribute should be removed.
-pluginattrname <*plugin custom attribute name*>	Specify the name of the custom attribute to be removed.
[-help]	*Optional*. Displays usage options for this command.

A.9.4 `removeplugincustomattr` Command

Use the `removeplugincustomattr` command to remove a custom attribute from a plug-in.

> **Note:** You must be an administrator with the Security Administrator Admin Role or the *Application Configurator* Admin Role to execute this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug]  -x removeplugincustomattr
```

The following table describes the options you can use with this command:

Option	Description
-pluginname <*plugin name*>	Identify the plug-in from which the custom attribute should be removed.
-pluginattrname <*plugin custom attribute name*>	Specify the name of the custom attribute to be removed.
[-help]	*Optional*. Displays usage options for this command.

A.9.5 `retrieveplugincustomattr` Command

Use the `retrieveplugincustomattr` command to retrieve a custom attribute from a plug-in.

> **Note:** You must be an administrator with the Security Administrator Admin Role or the *Application Configurator* Admin Role to execute this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x retrieveplugincustomattr
<options>
```

The following table describes the options you can use with this command:

Option	Description
-classname <plugin class name>	Identify the class name to retrieve from the plug-in.
-classpath <plugin class paths>	Identify the class path to retrieve from the plug-in
[-help]	*Optional*. Displays usage options for this command.

Note: You must specify all multi-valued attributes in this format: `value1|value2|...`

A.9.6 `searchplugin` Command

Use the `searchplugin` command to search for a plug-in.

Note: You must be an administrator with the Security Administrator Admin Role, the *User Manager* Admin Role, or the *Application Configurator* Admin Role to execute this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug]  -x searchplugin <options>
```

The following table describes the options you can use with this command:

Option	Description
[-pluginname] <*plugin name*>	*Optional*. Identify the plug-in to search for.
[-description] <*plugin description*>	*Optional*. Identify the plug-in description to search for.
[-pluginstatus] <*active/disabled*>	*Optional*. Identify the plug-in status to search for.
[-resource] <*target/account/server*>	*Optional*. Identify the plug-in resource to search for.
[-operation] <*plugin operation*>	*Optional*. Identify the plug-in operation to search for.
[-timing] <*pre/post*>	*Optional*. Identify the plug-in timing to search for.
[-help]	*Optional*. Displays usage options for this command.

You can use any combination of `-pluginname`, `-description`, `-pluginstatus`, `-resource`, `-operation` or `-timing` to identify the plug-in. If you do not provide any of these options, then the search returns all plug-ins.

For example, the following search returns all plug-ins:

```
https://<host name>:<port>/opam/plugin/search?
```

Whereas, the following search returns all plug-ins whose status is active and timing is pre:

```
https://<host name>:<port>/opam/plugin/search?pluginstatus=active&timing=pre
```

A.9.7 `modifyplugin` Command

Use the `modifyplugin` command to modify a plug-in.

Note: You must be an administrator with the *Security Administrator* Admin Role or the *Application Configurator* Admin Role to execute this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug]  -x modifyplugin <options>
```

The following table describes the options you can use with this command:

Note: You must specify all multi-valued attributes in this format: `value1|value2|...`

Option	Description
-pluginname <*plugin name*>	Identify the plug-in to be modified.
-propertyname <*propertyname*>	Specify the name of the property that you want to modify.
-propertyvalue <*propertyvalue*>	Specify the property value that you want to modify.
[-help]	*Optional*. Displays usage options for this command.

You can modify plug-in with the following property names:

Note: These property names are case-sensitive.

Property Name	Description
pluginStatus <*active/disabled*>	Modify the plug-in's status.
pluginDescription	Modify the plug-in description.
pluginResource <*target/account/server*>	Modify the resource on which the plug-in will perform.
pluginOperation	Modify the operation the plug-in performs.
pluginTiming <*pre/post*>	Modify the plug-in timing.
pluginOrder	Modify the plug-in order.
pluginClassName	Modify the plug-in's class name.
pluginClassPath [multi-valued]	Modify the plug-in's class path.
pluginEnableUser [multi-valued]	Modify the plug-in's enabled user list.
pluginDisableUser [multi-valued]	Modify the plug-in's disabled user list.
pluginEnableGroup [multi-valued]	Modify the plug-in's enabled group list.
pluginDisableGroup [multi-valued]	Modify the plug-in's disabled group list.
pluginEnableResourceGroup [multi-valued]	Modify the plug-in's enabled resource group list.
pluginDisableResourceGroup [multi-valued]	Modify the plug-in's disabled resource group list.
pluginEnableHTTPResult [multi-valued]	Modify the plug-in's enabled HTTP response.
pluginDisableHTTPResult [multi-valued]	Modify the plug-in's disabled HTTP response.
pluginVersion	Modify the plug-in's version.

Property Name	Description
pluginTimeout	Modify the plug-in's timeout.

A.9.8 `removeplugin` Command

Use the `removeplugin` command to remove a plug-in.

> **Note:** You must be an administrator with the *Application Configurator* Admin Role to execute this command.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug]  -x removeplugin <options>
```

The following table describes the options you can use with this command:

Option	Description
-pluginname <*plugin name*>	Identify the plug-in to be removed.
[-help]	*Optional*. Displays usage options for this command.

A.10 Exporting and Importing Data

The following sections contain information about the commands that you use when exporting and importing Oracle Privileged Account Manager data.

- Section A.10.1, "`export` Command"
- Section A.10.2, "`filedecryption` Command"
- Section A.10.3, "`import` Command"

A.10.1 `export` Command

Use the `export` command to export data stored in Oracle Privileged Account Manager, such as targets and accounts, to XML format. This option and the "`import` Command" on page A-57 are useful for performing the following operations:

- Bulk operations, such as querying or loading large volumes of data
- Back-up and recovery operations, such as periodically backing up Oracle Privileged Account Manager data to XML
- Migration operations, such as exporting data from one Oracle Privileged Account Manager instance and importing it to another instance

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to use these commands.

The `export` command exports all Oracle Privileged Account Manager data; including targets, accounts, policies, and grants.

Note: Exporting accounts also exports the passwords for those accounts. For added security, you can export the passwords in an encrypted format by using the `-encpassword` and `-enckeylen` options.

Be sure to note the encryption password and *encryption key length* because you must provide that same password for decryption during the `import` operation.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x export <options>
```

The following table describes the options you can use with the `export` command:

Option	Description
-f <*export file*>	Specify an export file name.
[-encpassword <*encryption password*>]	*Optional*. Specify a password to use when encrypting the account passwords to the exported file.
[-enckeylen <*key length for password encryption*>]	*Optional*. Specify the minimum key length for an encryption or decryption password. (Defaults to *128 bits*)
[-log <*log file location*>]	*Optional*. Specify a file name and location for the log file. (Defaults to `opamlog_<timestamp>.txt`)
[-noencrypt <*true/false*>]	*Optional*. Specify whether to provide an encryption password. (Defaults to *false*) ■ **true**: Skip the encryption password and export the output file in clear text. ■ **false**: Encrypt the output file with the encryption password.
[-help]	*Optional*. Displays usage options for this command.

The XML schema for an export file is located in the following file:

ORACLE_HOME/opam/jlib/OPAMBulkTool.xsd

The following example shows some sample XML definitions of Oracle Privileged Account Manager elements.

Example A–3 Sample XML Definition of Oracle Privileged Account Manager Elements

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<OPAMData xmlns="http://www.example.org/OPAMBulkTool"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.example.org/OPAMBulkTool OPAMBulkTool.xsd">
  <usagepolicy>
    <name value="Accounting Usage Policy"/>
    <status value="active"/>
    <description value="My Usage Policy"/>
    <globaldefault value="n"/>
    <dateorduration value="duration"/>
    <expiremin value="30"/>
    <expiredate value="08/08/2088"/>
    <expiretime value="11:30am"/>
    <timezone value="Etc/UTC"/>
    <allowcheckouttype value="all"/>
```

```
  <enablerecording value="true"/>
  <usagedays>
    <day fromtime="12:00am" totime="12:00am" value="monday"/>
    <day fromtime="12:00am" totime="12:00am" value="tuesday"/>
    <day fromtime="12:00am" totime="12:00am" value="wednesday"/>
    <day fromtime="12:00am" totime="12:00am" value="thursday"/>
    <day fromtime="12:00am" totime="12:00am" value="friday"/>
    <day fromtime="12:00am" totime="12:00am" value="saturday"/>
    <day fromtime="12:00am" totime="12:00am" value="sunday"/>
  </usagedays>
  <ssh enableInteractive="true" enableNonInteractive="true">
    <commandControl listType="blacklist" listValues="rm"/>
  </ssh>
  <scp enable="true"/>
</usagepolicy>
<passwordpolicy>
  <name value="Accounting Password Policy"/>
  <status value="active"/>
  <description value=""/>
  <globaldefault value="n"/>
  <changepassevery value="30-days"/>
  <changepasscheckout value="y"/>
  <changepasscheckin value="y"/>
  <passwordlength max="20" min="8"/>
  <minalphabets value="1"/>
  <minnumeric value="1"/>
  <minalphanumeric value="2"/>
  <specialchars max="5" min="1"/>
  <repeatedchars max="1" min="0"/>
  <minuniquechars value="1"/>
  <minuppercasechars value="1"/>
  <minlowercasechars value="1"/>
  <startwithchar value="n"/>
  <accountnameaspass value="n"/>
  <passwordhistorydays value="30"/>
</passwordpolicy>
<connectorserver>
  <name value="USConnectorServer"/>
  <host value="hostname"/>
  <port value="portnumber"/>
  <sslenabled value="false"/>
  <key value="keyvalue"/>
  <timeout value="60"/>
</connectorserver>
<plugin>
  <name value="OPAMPasswordSyncPlugin"/>
  <resource value="account"/>
  <operation value="add"/>
  <timing value="post"/>
  <order value="1"/>
  <classname value="OPAMPasswordSyncPlugin"/>
  <classpath value="/plugins/OPAMPasswordSyncPlugin.jar"/>
  <status value="active"/>
  <timeout value="60"/>
  <enableuser value="opam_user"/>
  <enableresourcegroup value="US_Resources"/>
  <enablehttpresult value="200"/>
  <customattributes>
    <customattribute attributename="email" attributevalue="notification_email@hostname.com"/>
    <customattribute attributename="tag" attributevalue="Internal,Password"/>
```

```
    </customattributes>
  </plugin>
  <target>
    <type name="database"/>
    <name value="AccountsDB"/>
    <attributes>
      <attributeName name="host" value="hostname"/>
      <attributeName name="jdbcUrl" value="jdbc:oracle:thin:@hostname:portnumber:servicename"/>
      <attributeName name="loginUser" value="login"/>
      <attributeName name="loginPassword" value="password"/>
      <attributeName name="dbType" value="Oracle"/>
      <attributeName name="description" value=""/>
      <attributeName name="domain" value="Accounting"/>
      <attributeName name="passwordpolicy" value="Accounting Password Policy"/>
      <attributeName name="connectorserverid" value=""/>
      <attributeName name="passwordrollover" value="false"/>
      <attributeName name="connectionProperties" value=""/>
    </attributes>
    <customattributes>
      <customattribute attributename="Owner" attributevalue="jack@company.com,tom@company.com"/>
    </customattributes>
  </target>
  <account>
    <name value="ACCT_DBA"/>
    <target name="AccountsDB"/>
    <description value=""/>
    <passwordpolicy name="Accounting Password Policy"/>
    <grantee>
      <user name="johndoe" usagepolicy="Accounting Usage Policy"/>
      <user name="janedoe" usagepolicy="Default Usage Policy"/>
    </grantee>
    <shared value="false"/>
    <status value="checkedIn"/>
    <keyMaps/>
    <customattributes>
      <customattribute attributename="PasswordSyncRequired" attributevalue="True"/>
    </customattributes>
  </account>
  <resourcegroup>
    <name value="US_Resources"/>
    <members>
      <member memberName="[AccountsDB][ACCT_DBA]" memberType="account"/>
      <member memberName="AccountsDB" memberType="target"/>
    </members>
    <delegations>
      <delegation delegatee="US_admin" delegatee_type="user" privilege="security_admin"/>
    </delegations>
  </resourcegroup>
</OPAMData>
```

A.10.2 `filedecryption` Command

Use the `filedecryption` command to decrypt an encrypted Oracle Privileged Account Manager configuration file.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x filedecryption
-f <encrypted file> -df <destination file> [-encpassword <decryption password>]
<options>
```

Note: This operation does not require any server connectivity when the `-offline true` option is provided. When using `-offline true` you can omit providing `-url, -u` and `-p` options. Files exported using Oracle Privileged Account Manager version 11.1.2.2.0 or later can be decrypted using `-offline true` option. Files exported using older versions of Oracle Privileged Account Manager version 11.1.2.2.0 cannot be decrypted offline.

The following table describes the options you can use with this command:

Option	Description
-f <*file with encrypted data*>	Specify the encrypted Oracle Privileged Account Manager configuration file.
-df <*file to write decrypted data*>	Specify where to write the decrypted file.
[-encpassword <*encryption/decryption password*>]	*Optional*. Specify the password to use when decrypting the data.
[-enckeylen <*Key length for encryption/decryption password*>]	*Optional*. Specify the minimum key length for an encryption/decryption password. (Defaults to *128 bits*)
[-force <*true/false*>]	*Optional*. Enables or disables the requirement for connection validation. ■ **true**: Skips connection validation. ■ **false** (*default*): Enforces connection validation.
[-log <*log file location*>]	*Optional*. Specify a file name and location for the log file. (Defaults to `opamlog_<timestamp>.txt`)
[-offline <*true/false*>]	Specify whether the command can connect to the Oracle Privileged Account Manager server. ■ **true**: Command will not connect to the server. ■ **false** (*default*): Command will connect to the server.
[-help]	*Optional*. Displays usage options for this command.

For example, use the following command if you do not have server connectivity:

```
sh opam.sh -x filedecryption -f <encrypted file> -df <destination file>
  -offline true
```

A.10.3 `import` Command

Use the `import` command to import data to Oracle Privileged Account Manager from an XML file. This option and the "`export` Command" on page A-53 are useful for performing the following operations:

- Bulk operations, such as querying or loading large volumes of data
- Back-up and recovery operations, such as periodically backing up Oracle Privileged Account Manager data to XML
- Migration operations, such as exporting data from one Oracle Privileged Account Manager instance and importing it to another instance

Note: You must be an administrator with both the *Security Administrator* Admin Role and the *User Manager* Admin Role to use these commands.

If the account status is checked-in, users do not have to provide status when importing data to Oracle Privileged Account Manager.

You can create an import XML file from previously exported data or you can manually create the file. If you previously exported the XML file with an encryption password, then you must provide the same password for decryption during import.

In addition to object creation, you can also use the `import` command to update and delete objects. Refer to reference for more information.

Command Syntax:

```
[-url <url>] -u <username> [-p <password>] [-debug] -x import <options>
```

The following table describes the options you can use with this command:

Option	Description
-f <*import file*>	Specify an import file name.
[-encpassword <*encryption password*>]	*Optional*. Specify a password to use when decrypting account passwords from the exported file.
[-enckeylen <*key length for password encryption*>]	*Optional*. Specify the minimum key length for an encryption/decryption password. (Defaults to *128 bits*)
[-force <*true/false*>]	*Optional*. Enables or disables the requirement for connection validation. ■ **true**: Skips connection validation. ■ **false** (*default*): Enforces connection validation.
[-log <*log file location*>]	*Optional*. Specify a file name and location for the log file. (Defaults to `opamlog_<timestamp>.txt`)
[-noencrypt <*true/false*>]	*Optional*. Specify whether to decrypt the imported file. (Defaults to *false*) ■ **true**: Skip the encryption password. The system will import the file in clear text. ■ **false**: Use the encryption password to decrypt the import file, and then load the decrypted data into the system.
[-help]	*Optional*. Displays usage options for this command.

The XML schema for an import file is located in the following file:

```
ORACLE_HOME/opam/jlib/OPAMBulkTool.xsd
```

The following examples show some sample XML definitions of Oracle Privileged Account Manager elements.

Example A–4 Data Creation

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<OPAMData xmlns="http://www.example.org/OPAMBulkTool"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.example.org/OPAMBulkTool OPAMBulkTool.xsd">
  <usagepolicy>
```

```
  <name value="Accounting Usage Policy"/>
  <status value="active"/>
  <description value="My Usage Policy"/>
  <globaldefault value="n"/>
  <dateorduration value="duration"/>
  <expiremin value="30"/>
  <expiredate value="08/08/2088"/>
  <expiretime value="11:30am"/>
  <timezone value="Etc/UTC"/>
  <allowcheckouttype value="all"/>
  <enablerecording value="true"/>
  <usagedays>
    <day fromtime="12:00am" totime="12:00am" value="monday"/>
    <day fromtime="12:00am" totime="12:00am" value="tuesday"/>
    <day fromtime="12:00am" totime="12:00am" value="wednesday"/>
    <day fromtime="12:00am" totime="12:00am" value="thursday"/>
    <day fromtime="12:00am" totime="12:00am" value="friday"/>
    <day fromtime="12:00am" totime="12:00am" value="saturday"/>
    <day fromtime="12:00am" totime="12:00am" value="sunday"/>
  </usagedays>
  <ssh enableInteractive="true" enableNonInteractive="true">
    <commandControl listType="blacklist" listValues="rm"/>
  </ssh>
  <scp enable="true"/>
</usagepolicy>
<passwordpolicy>
  <name value="Accounting Password Policy"/>
  <status value="active"/>
  <description value=""/>
  <globaldefault value="n"/>
  <changepassevery value="30-days"/>
  <changepasscheckout value="y"/>
  <changepasscheckin value="y"/>
  <passwordlength max="20" min="8"/>
  <minalphabets value="1"/>
  <minnumeric value="1"/>
  <minalphanumeric value="2"/>
  <specialchars max="5" min="1"/>
  <repeatedchars max="1" min="0"/>
  <minuniquechars value="1"/>
  <minuppercasechars value="1"/>
  <minlowercasechars value="1"/>
  <startwithchar value="n"/>
  <accountnameaspass value="n"/>
  <passwordhistorydays value="30"/>
</passwordpolicy>
<connectorserver>
  <name value="USConnectorServer"/>
  <host value="hostname"/>
  <port value="portnumber"/>
  <sslenabled value="false"/>
  <key value="keyvalue"/>
  <timeout value="60"/>
</connectorserver>
<plugin>
  <name value="OPAMPasswordSyncPlugin"/>
  <resource value="account"/>
  <operation value="add"/>
  <timing value="post"/>
  <order value="1"/>
```

```
    <classname value="OPAMPasswordSyncPlugin"/>
    <classpath value="/plugins/OPAMPasswordSyncPlugin.jar"/>
    <status value="active"/>
    <timeout value="60"/>
    <enableuser value="opam_user"/>
    <enableresourcegroup value="US_Resources"/>
    <enablehttpresult value="200"/>
    <customattributes>
      <customattribute attributename="email"
attributevalue="notification_email@hostname.com"/>
      <customattribute attributename="tag" attributevalue="Internal,Password"/>
    </customattributes>
  </plugin>
  <target>
    <type name="database"/>
    <name value="AccountsDB"/>
    <attributes>
      <attributeName name="host" value="hostname"/>
      <attributeName name="jdbcUrl"
value="jdbc:oracle:thin:@hostname:portnumber:servicename"/>
      <attributeName name="loginUser" value="login"/>
      <attributeName name="loginPassword" value="password"/>
      <attributeName name="dbType" value="Oracle"/>
      <attributeName name="description" value=""/>
      <attributeName name="domain" value="Accounting"/>
      <attributeName name="passwordpolicy" value="Accounting Password Policy"/>
      <attributeName name="connectorserverid" value=""/>
      <attributeName name="passwordrollover" value="false"/>
      <attributeName name="connectionProperties" value=""/>
    </attributes>
    <customattributes>
      <customattribute attributename="Owner"
attributevalue="jack@company.com,tom@company.com"/>
    </customattributes>
  </target>
  <account>
    <name value="ACCT_DBA"/>
    <target name="AccountsDB"/>
    <description value=""/>
    <passwordpolicy name="Accounting Password Policy"/>
    <grantee>
      <user name="johndoe" usagepolicy="Accounting Usage Policy"/>
      <user name="janedoe" usagepolicy="Default Usage Policy"/>
    </grantee>
    <shared value="false"/>
    <status value="checkedIn"/>
    <keyMaps/>
    <customattributes>
      <customattribute attributename="PasswordSyncRequired"
attributevalue="True"/>
    </customattributes>
  </account>
  <resourcegroup>
    <name value="US_Resources"/>
    <members>
      <member memberName="[AccountsDB][ACCT_DBA]" memberType="account"/>
      <member memberName="AccountsDB" memberType="target"/>
    </members>
    <delegations>
      <delegation delegatee="US_admin" delegatee_type="user"
```

```
privilege="security_admin"/>
    </delegations>
  </resourcegroup>
</OPAMData>
```

Example A–5 Data Modification: Modify An Account Password Policy

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<OPAMData xmlns="http://www.example.org/OPAMBulkTool"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.example.org/OPAMBulkTool OPAMBulkTool.xsd">
  <account operation="modify">
    <name value="account2"/>
    <target name="lockbox_target1"/>
    <passwordpolicy name="test-pass-policy"/>
    <shared value="true"/>
  </account>
</OPAMData>
```

Example A–6 Data Modification: Modify A Password Policy

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<OPAMData xmlns="http://www.example.org/OPAMBulkTool"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.example.org/OPAMBulkTool OPAMBulkTool.xsd">
<passwordpolicy operation="modify">
    <name value="test policy"/>
    <status value="active"/>
    <description value="test"/>
    <globaldefault value="n"/>
    <changepassevery value="45-hours"/>
    <changepasscheckout value="n"/>
    <changepasscheckin value="n"/>
    <passwordlength max="20" min="5"/>
    <minalphabets value="0"/>
    <minnumeric value="0"/>
    <minalphanumeric value="0"/>
    <specialchars max="5" min="0"/>
    <repeatedchars max="10" min="0"/>
    <minuniquechars value="0"/>
    <minuppercasechars value="0"/>
    <minlowercasechars value="0"/>
    <startwithchar value="y"/>
    <requiredchars value="a,b,c,d,e"/>
    <allowedchars value="a,b,c,d,e,f,g,h"/>
    <disallowedchars value="z,-,x"/>
    <accountnameaspass value="y"/>
  </passwordpolicy>
</OPAMData>
```

Example A–7 Data Deletion: Delete a Target

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<OPAMData xmlns="http://www.example.org/OPAMBulkTool"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.example.org/OPAMBulkTool OPAMBulkTool.xsd">
  <target operation="delete">
    <type name="lockbox"/>
```

```
    <name value="lockbox_target1"/>
  </target>
</OPAMData>
```

Example A–8 Data Deletion: Delete an Account

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<OPAMData xmlns="http://www.example.org/OPAMBulkTool"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.example.org/OPAMBulkTool OPAMBulkTool.xsd">
<account operation="delete">
    <name value="account3"/>
    <target name="lockbox_target1"/>
</account>
<account operation="delete">
    <name value="account4"/>
    <target name="lockbox_target1"/>
</account>
</OPAMData>
```

B

Working with Oracle Privileged Account Manager's RESTful Interface

This appendix describes Oracle Privileged Account Manager's RESTful interface, including the specific APIs that are exposed through this interface.

This appendix includes the following sections:

B.1 Overview

While Oracle Privileged Account Manager can be consumed through several client interfaces, its fundamental access mechanism or layer is encapsulated in its RESTful interfaces.

> **Note:** For information about using Oracle Privileged Account Manager's web-based Console or command line tool to perform tasks described in this appendix, refer to Chapter 4, "Starting and Using the Oracle Privileged Account Manager Console" or Appendix A, "Working with the Command Line Tool."

All interactions with Oracle Privileged Account Manager's server that are being used by external parties, such as a non-Oracle Privileged Account Manager server, are exposed through RESTful interfaces. All externally visible Oracle Privileged Account

Manager resources are modeled by URIs, while standard HTTP operations are mapped to relevant Oracle Privileged Account Manager operations on those resources.

Note: The information provided in this appendix is essentially the same whether you are using Oracle Privileged Account Manager on WebLogic or on IBM WebSphere; however, there are a few minor differences.

For more information, refer to "Differences When Using the Oracle Privileged Account Manager Command Line Tool and REST Interfaces on IBM WebSphere" in the *Oracle Fusion Middleware Third-Party Application Server Guide for Oracle Identity and Access Management*.

B.2 Server State Resource

This section describes the Get Server State API.

B.2.1 Get Server State

Use this API to retrieve information about the status of a server.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/
- **Method:** `GET`
- **Content-Type:** NA
- **Returns on Success:** Status code 200 and JSON representation of the Server State Resource

Example B–1 Sample JSON Output of Server Status

```
{
    "RequestorGroups" : [
        "OPAM_APPLICATION_CONFIGURATOR",
        "OPAM_SECURITY_ADMIN",
        "OPAM_USER_MANAGER",
        "OPAM_SECURITY_AUDITOR"
    ],
    "ServerState" : {
        "Status" : "Oracle Privileged Account Manager Server is up!",
        "StatusCode" : 0
    },
    "Requestor" : "master_user"
    "version":"11.1.2.2.0"
}

{
    "ServerState" : {
        "Status" : "Oracle Privileged Account Manager Server is up!",
        "StatusCode" : 0
    },
    "Requestor" : "sec_admin",
    "RequestorGroups" : [
    "OPAM_SECURITY_ADMIN"
    ],
    version: "11.1.2.3.0",
    DelegatedPrivileges: [ ]
```

```
}

{
   "ServerState":{
      "Status":"Oracle Privileged Account Manager Server is up!",
      "StatusCode":0
   },
   "Requestor":"weblogic",
   "RequestorGroups":[
      "Administrators"
   ],
   "version":"11.1.2.3.0",
   "DelegatedPrivileges":[
      "security_admin"
   ]
}
```

Where:

- **RequestorGroups** are groups assigned to the user who is making the request.
- **Requestor** is the user who is making the request.
- **StatusCode** indicates whether the server is working properly.
  - Returns a zero (**0**) if the server is working properly.
  - Returns a non-zero integral value if the server has encountered some issue.
- **Status** is an informative message about the state of the server.
- **version** is the Oracle Privileged Account Manager version.
- **DelegatedPrivileges** are the administration privileges delegated to the user. If the user was delegated an admin privilege such as "security_admin" or "user_manager" on a resource group, the privilege will be displayed in this attribute.

  If no admin privilege was delegated to the user, the attribute will remain empty as shown in the second sample JSON output in Example B–1.

B.3 Connector Server Configuration Resource

This section describes the following configuration resource APIs:

- Section B.3.1, "Add Connector Server Configuration"
- Section B.3.2, "Verify a Connector Server Configuration"
- Section B.3.3, "Update Connector Server Configuration"
- Section B.3.4, "Delete Connector Server Configuration"
- Section B.3.5, "Get Connector Server Configuration"
- Section B.3.6, "Search Connector Server Configuration"

B.3.1 Add Connector Server Configuration

Use this API to add a connector server configuration.

Note: You must be an administrator with the "Application Configurator" Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/connectorserver
- **Method:** `POST`
- **Content-Type:** application/json
- **Body:** JSON representation of connector server for addition/test
- **Returns on Success:** Status code 201 Created and Location

Example B–2 Sample JSON Representation of Connector Server Configuration for Addition

```
{
 "connectorserver": {
   "connectorservername": "server_test",
   "connectorserverdescription": "demo connector server in US",
   "connectorserverhost": "myhost.us.example.com",
   "connectorserverport": 8579,
   "connectorserverkey": "password2",
   "connectorserversslenabled ": true
 }
}
```

Sample Output:

```
https://opam_server_host:opam_ssl_port/opam/connectorserver/9bbcbbb087174ad1900ea6
91a2573b61 as the Location
```

Where:

- **connectorservername** is the name given to the connector server.
- **connectorserverdescription** is the description for the connector server. It is an optional field.
- **connectorserverhost** is the hostname of the connector server.
- **connectorserverport** is the port of the connector server.
- **connectorserverkey** is the key of the connector server.
- **connectorserversslenabled** indicates whether SSL is enabled on the connector server.

B.3.2 Verify a Connector Server Configuration

Use this API to verify a connector server configuration before addition.

> **Note:** You must be an administrator with the "Application Configurator" Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/connectorserver/test
- **Method:** `POST`
- **Content-Type:** application/json
- **Body:** JSON representation of connector server for addition/test
- **Returns on Success:** Status code 200

Example B–3 Sample JSON Representation of Connector Server Configuration for Addition

```
{
 "connectorserver": {
```

```
   "connectorservername": "server_test",
   "connectorserverdescription": "demo connector server in US",
   "connectorserverhost": "myhost.us.example.com",
   "connectorserverport": 8579,
   "connectorserverkey": "password2",
   "connectorserversslenabled ": true
 }
}
```

Where:

- **connectorservername** is the name given to the connector server.
- **connectorserverdescription** is the description for the connector server. It is an optional field.
- **connectorserverhost** is the hostname of the connector server.
- **connectorserverport** is the port of the connector server.
- **connectorserverkey** is the key of the connector server.
- **connectorserversslenabled** indicates whether SSL is enabled on the connector server.

B.3.3 Update Connector Server Configuration

Use this API to update a connector server configuration.

> **Note:** You must be an administrator with the "Application Configurator" Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/connectorserver/*connector_server_id*
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation of connector server modification
- **Returns on Success:** Status code 200

Example B–4 Sample JSON Representation of Connector Server Configuration Modification

```
{
"modifications":[
   {
       "modification": {
           "connectorserverhost": "myhost.us.example.com"
       }
   },
       {
       "modification": {
           "connectorserverport":8670
       }
   }
   ]
}
```

Where:

- **connectorserverhost** is the hostname of the connector server.

- **connectorserverport** is the port of the connector server.

B.3.4 Delete Connector Server Configuration

Use this API to delete a connector server configuration.

> **Note:** You must be an administrator with the "Application Configurator" Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/connectorserver/*connector_server_id*
- **Method:** `DELETE`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200

B.3.5 Get Connector Server Configuration

Use this API to retrieve a connector server configuration.

> **Note:** You must be an administrator with the "Application Configurator," "Security Administrator," or "Delegate Security Administrator" Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/connectorserver/*connector_server_id*
- **Method:** `GET`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON Representation of Connector Server

Sample Output:

```
{
 "connectorserver": {
   "connectorserverid": "ab62a4b85ba34c9499794ab181d37c15",
   "connectorservername": "server_test",
   "connectorserverdescription": "demo connector server in US",
   "connectorserverhost": "myhost.us.example.com",
   "connectorserverport": 8579,
   "connectorserverkey": "password2",
   "connectorserversslenabled": true
 }
}
```

Where:

B.3.6 Search Connector Server Configuration

Use this API to search connector server configurations.

> **Note:** You must be an administrator with the "Application Configurator," "Security Administrator," or "Delegate Security Administrator" Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/connectorserver//search?param1=val1&param2=val2..
- **Method:** `GET`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON Representation of Connector Server

Where query parameters could be host, name, and description.

Sample Query:

```
https://myhost.example.com:2001/opam/connectorserver/search?name=server&host=oracl
e
```

Sample Query:

```
{
 "connectorservercollection": [
   {
     "connectorserver": {
       "connectorserverid": "21ae721b54854b3790214fd3fa6864df",
       "connectorservername": "server1",
       "connectorserverhost": "myhost.us.example.com",
       "connectorserverport": 180,
       "connectorserversslenabled ": true
     }
   },
   {
     "connectorserver": {
       "connectorserverid": "0a24c6287aa44d6a814b6f4deb7b751b",
       "connectorservername": "server2",
       "connectorserverhost": "myhost.us.example.com",
       "connectorserverport": 280,
       "connectorserversslenabled ": true

     }
   },
   {
     "connectorserver": {
       "connectorserverid": "ab62a4b85ba34c9499794ab181d37c15",
       "connectorservername": "server3",
       "connectorserverdescription": "optional description",
       "connectorserverhost": "myhost.us.example.com",
       "connectorserverport": 45,
       "connectorserversslenabled ": true
```

```
      }
    }
  ]
}
```

Where:

- **connectorserverid** is the ID of the connector server in your environment.
- **connectorservername** is the name given to the connector server.
- **connectorserverdescription** is the description for the connector server. It is an optional field.
- **connectorserverhost** is the hostname of the connector server.
- **connectorserverport** is the port of the connector server.
- **connectorserversslenabled** indicates whether SSL is enabled on the connector server.

B.4 Configuration Resource

This section describes the following configuration resource APIs:

- Section B.4.1, "Global Configuration Resource"
- Section B.4.2, "Oracle Privileged Session Manager Configuration Resource"

B.4.1 Global Configuration Resource

The APIs described in this section include:

- Section B.4.1.1, "Get Configuration Resource"
- Section B.4.1.2, "Update Configuration Resource"

B.4.1.1 Get Configuration Resource

Use this API to retrieve a configuration object for Oracle Privileged Account Manager.

> **Note:** You must be an administrator with the *User Manager* Admin Role, the *Security Administrator* Admin Role, or the *Application Configurator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/config/*configid*
- **Method:** `GET`
- **Content-Type:** NA
- **Returns on Success:** Status code 200 and JSON representation of a `config` object

Sample URI

https://*opam_server_host*:*opam_ssl_port*/opam/config/*globalconfig*

Example B–5 Sample JSON Representation of a config Object

```
{
config: {
configUID: "globalconfig",
```

```
configType: "config_globalconfig",
idstorefilter: [
"beginswith"
],
tdemode: [
"true"
],
resourceLockWaitTimeout: [
"120"
],
policyenforcerinterval: [
"3600"
],
targettimeout: [
"20"
],
passwddisplayoption: [
"showpasswd"
],
passwordcyclerinterval: [
"3600"
]
}
}
```

Where:

- **configUID** is a unique identifier for the `config` object.
- **configType** is the type of `config` object.
- **policyenforcerinterval** is the interval (in seconds) in which Oracle Privileged Account Manager checks accounts and then automatically checks-in the accounts that have exceeded the expiration time defined in the Usage Policy.
- **passwordcyclerinterval** is the interval (in seconds) in which Oracle Privileged Account Manager checks and then resets the password for any accounts that have exceeded the maximum password age defined in the Password Policy.
- **passworddisplayoption** determines how Oracle Privileged Account Manager displays the password of an account to the user.
- **targettimeout** is the time (in seconds) allowed to perform the target connectivity test operation.
- **tdemode** is a flag to request that Oracle Privileged Account Manager use TDE or non-TDE mode.
- **resourceLockWaitTimeout** is the Maximum time (in seconds) allowed for an operation to obtain a transaction lock on a resource.

B.4.1.2 Update Configuration Resource

Use this API to modify a configuration object for Oracle Privileged Account Manager.

> **Note:** You must be an administrator with the *Application Configurator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/config/*configid*
- **Method:** `PUT`

- **Content-Type:** application/json
- **Body:** JSON representation of Modification
- **Returns on Success:** Status code 200

Example B–6 Sample JSON Output of Modification

```
{
    "modifications":[
        {
            "modification":{
                "tdemode":[
                    "false"
                ]
            }
        }
    ]
}
```

Where:

- **modifications** are an array of modification JSON objects.
- **modification** is a JSON object representing the modification of a single configuration object.
- **tdemode** is a flag to request that Oracle Privileged Account Manager use TDE or non-TDE mode.

B.4.2 Oracle Privileged Session Manager Configuration Resource

The APIs described in this section include:

- Section B.4.2.1, "Get Configuration Resource"
- Section B.4.2.2, "Update Configuration Resource"

B.4.2.1 Get Configuration Resource

Use this API to get a configuration object for Oracle Privileged Session Manager.

Note:

- You must be an administrator with the *User Manager*, the *Security Administrator*, or the *Application Configurator* Admin Role to use this API.
- You cannot run two instances of Oracle Privileged Session Manager on the same machine.

- **URI**: https://opam_server_host:opam_ssl_port/opam/config/sessionmgrconfig
- **Method:** `GET`
- **Content-Type:** NA
- **Returns on Success:** Status code 200 and JSON Representation of a Session Manager `config` object

Example B–7 Sample JSON Representation of Session Manager Config

```
 {
config: {
updateinterval: 60,
maxauditthreads: 5,
maxsessions: 8192,
maxrecordsize: 10240,
restResponseTimeout: 180,
SSH: {
opamListenPort: 1222,
sessionchkoutinstructions: "ssh -p <port> <opamuser>:<targetname>:<accountname>@<sessionmgrhost>
Use opam password on password prompt"
},
SMVS: {
opamListenPort: 5389,
imagestorepath: "/scratch/opam/images"
},
configUID: "sessionmgrconfig",
configType: "config_sessionmgrconfig",
windowsAgentCount: 0
}
}
```

Where:

- **configUID** is a unique identifier for the `config` object.
- **configType** is the type of `config` object.
- **updateinterval** is the interval (in seconds) in which the Oracle Privileged Session Manager server checks all of the checked-out sessions and updates their transcripts.
- **opamserverurls** is an array of Oracle Privileged Account Manager server URLs to which Oracle Privileged Session Manager can connect.
- **pub-key** is the Oracle Privileged Session Manager server's public key.
- **maxrecordsize** is the maximum recording size that is allowed per session (in KB). When this quota is reached, the session is automatically terminated.
- **prv-key** is the Oracle Privileged Session Manager server's private key.

Protocol-specific attributes include:

- **opamListenPort** is the listener port for the protocol.
- **sessionchkoutinstructions** is the session checkout instructions.
- `restResponseTimeout` is the maximum time allowed for Oracle Privileged Session Manager to complete Oracle Privileged Account Manager Server REST URL invocation.
- **maxauditthreads** is the maximum number of audit threads in the session manager audit pool.
- **maxsessions** is the maximum number of concurrent sessions allowed per session manager server.
- **windowAgentCount** is the number of windows agents that are deployed.

B.4.2.2 Update Configuration Resource

Use this API to update a configuration object for Oracle Privileged Session Manager.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/config/sessionmgrconfig
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation of Modification
- **Returns on Success:** Status code 200

Note: You must be an administrator with the *Application Configurator* Admin Role to use this API.

Example B–8 Sample JSON Modification

```
{
"modifications": [
{
"modification": {
"updateinterval": 300
}
},
{
"modification": {
"opamserverurls": [
"https://localhost:7002/opam"
]
}
},
{
"modification": {
"SSH": {
"opamListenPort": 1222
}
}
},
{
"modification": {
"SSH": {
"sessionchkoutinstructions":"ssh -p <port>
<opamuser>:<targetname>:<accountname>@<sessionmgrhost> \n Use opam password on password prompt"
}
}
}
]
}
```

Note: You can update all of these attributes, except

- **configUID** is a unique identifier for the `config` object.
- **configType** is the type of `config` object.
- **WindowsAgentCount** is the number of windows agents that have been deployed.

For the other attribute definitions, refer to Section B.4.2.1, "Get Configuration Resource."

B.5 Policy Resource

This section describes the APIs you use when working with Oracle Privileged Account Manager policies.

The APIs described in this section include:

- Section B.5.1, "Search for Policies"
- Section B.5.2, "Get Default Policies"
- Section B.5.3, "Password Policy Resource"
- Section B.5.4, "Usage Policy Resource"

B.5.1 Search for Policies

Use this API to search for policies. This API is a search, using one or more of the following parameters:

- `policystatus`
- `policyname`

All of the parameters are *optional*.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/policy/search?param1=val1&param2=val2
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of policies

Example B–9 Sample JSON Representation of Policies

```
{
   "usagepolicies":[
      {
         "policyname":"Default Usage Policy",
         "policyid":"usagepolicy1",
         "policystatus":"active",
      }
   ],
   "passwordpolicies":[
      {
         "policyname":"Default Password Policy",
         "policyid":"passwordpolicy2",
         "policystatus":"active",
         "globaldefault":"y"
      }
   ]
}
```

Where:

- **usagepolicies** are an array of Usage Policies.
- **passwordpolicies** are an array of Password Policies.
- **policyname** is the policy name.

- **policyid** is the policy's unique identifier.
- **policystatus** is the policy status, where acceptable values are `active` or `disabled`.

B.5.2 Get Default Policies

Use this API to get the Default Usage Policy and Default Password Policy.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/policy/default
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON Representation of policies

Example B–10 Sample JSON Representation of Policies

```
{
   "usagepolicies":[
      {
         "policyname":"Default Usage Policy",
         "policyid":"usagepolicy1",
         "policystatus":"active"
      }
   ],
   "passwordpolicies":[
      {
         "policyname":"Default Password Policy",
         "policyid":"passwordpolicy2",
         "policystatus":"active"
      }
   ]
}
```

Where:

- **usagepolicies** is an array of Usage Policies.
- **passwordpolicies** is an array of Password Policies.
- **policyname** is the policy name.
- **policyid** is the policy's unique identifier.
- **policystatus** is the policy status, where acceptable values are `active` or `disabled`.

This attribute only returns the default policies, Default Usage Policy and Default Password Policy.

B.5.3 Password Policy Resource

The APIs described in this section include:

- Section B.5.3.1, "Retrieve a Password Policy"
- Section B.5.3.2, "Update a Password Policy"
- Section B.5.3.3, "Create a Password Policy"
- Section B.5.3.4, "Get Accounts for Password Policy"
- Section B.5.3.5, "Delete a Password Policy"

B.5.3.1 Retrieve a Password Policy

Use this API to retrieve a Password Policy.

> **Note:** You must be an administrator with the *User Manager* Admin Role or the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/passwordpolicy/{policyid}
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of Password Policy

Example B–11 Sample JSON Representation of Password Policy

```
{
   "passwordpolicy":{
      "policyid":"passwordpolicy2",
      "policystatus":"active",
      "policyname":"Default Password Policy",
      "description":"Default Password Policy",
      "globaldefault":"y",
      "passwordchangedurationunit":"days",
      "passwordchangedurationvalue":30,
      "passwordhistorydays":30
      "changeoncheckin":"y",
      "changeoncheckout":"y",
      "passwordcharsmin":8,
      "passwordcharsmax":8,
      "passwordalphabeticmin":1,
      "passwordnumericmin":1,
      "passwordalphanumericmin":2,
      "passworduniquemin":1,
      "passworduppercasemin":1,
      "passwordlowercasemin":1,
      "passwordspecialmin":0,
      "passwordspecialmax":0,
      "passwordrepeatedmin":0,
      "passwordrepeatedmax":1,
      "startingchar":"n",
      "isaccountnameallowed":"n",
      "requiredchars":[
         "a",
         "h",
         "j"
      ],
      "allowedchars":[
         "b",
         "t",
         "y",
         "p",
         "u",
         "r",
         "o",
         "k",
         "1",
         "2",
```

```
            "=",
            "M",
            "a",
            "h",
            "j"
         ],
         "disalloweddchars":[
            "7",
            "8",
            "l"
         ],
      }
}
```

Where:

- **passwordpolicy** is a passwordpolicy JSON object.
- **policyid** is the policy's unique identifier.
- **policystatus** is the policy's status, where acceptable values are `active` or `disabled`.
- **policyname** is the policy name.
- **description** is a description of the policy.
- **globaldefault** indicates whether the policy is a global default or not.
- **passwordchangedurationunit** and **passwordchangedurationvalue** determine the interval after which the account password must be changed. Where `passwordchangedurationunit` can have the values: `days`, `hours`, or `minutes`.
- **passwordhistorydays** indicates how many days to keep the password history.
- **changeoncheckin** indicates whether to change the password on check-in. The valid values are `y` and `n`.
- **changeoncheckout** indicates whether to change the password on checkout. The valid values are `y` and `n`.
- **startingchar** indicates the character with which the password should begin.
- **isaccountnameallowed** indicates whether the password can be the same as the account name.
- **requiredchars**, **allowedchars**, **disallowedchars** are characters that are required, allowed, and disallowed respectively.
- **passwordcharsmin** is the minimum number of characters required in the password.
- **passwordcharsmax** is the maximum number of characters allowed in the password.
- **passwordalphabeticmin** is the minimum number of alphabetic characters required in the password.
- **passwordnumericmin** is the minimum number of numeric characters required in the password.
- **passwordalphanumericmin** is the minimum number of alphanumeric characters required in the password.
- **passworduniquemin** is the minimum number of unique characters required in the password.

- **passworduppercasemin** is the minimum number of uppercase characters required in the password.
- **passwordlowercasemin** is the minimum number of lowercase characters required in the password.
- **passwordspecialmin** is the minimum number of special characters required in the password.
- **passwordspecialmax** is the maximum number of special characters allowed in the password.
- **passwordrepeatedmin** is the minimum number of repeated characters required in the password.
- **passwordrepeatedmax** is the maximum number of repeated characters allowed in the password.

B.5.3.2 Update a Password Policy

Use this API to update a Usage Policy. You can update all of the attributes, except `policyid`, and you can update multiple attributes at a time.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/passwordpolicy/{policyid}
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation for Password Policy modification
- **Returns on Success:** Status code 200

Example B–12 Sample JSON Representation of Password Policy Modification

```
{
   "modifications":[
      {
         "modification":{
            "disalloweddchars":[
               "4",
               "6"
            ]
         }
      },
      {
         "modification":{
            "passwordalphabeticmin":2
         }
      }
   ]
}
```

Where:

- **modifications** is an array of modification JSON objects.
- **modification** is a JSON object representing a single attribute.

B.5.3.3 Create a Password Policy

Use this API to create a Password Policy.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/passwordpolicy
- **Method:** `POST`
- **Content-Type:** application/json
- **Body:** JSON representation for Password Policy creation
- **Returns on Success:** Status code 201

Example B–13 Sample JSON Representation for Password Policy Creation

```
{
   "passwordpolicy":{
      "policystatus":"active",
      "policyname":"Custom Password Policy",
      "description":"Default Password Policy",
      "passwordchangedurationunit":"days",
      "passwordchangedurationvalue":30,
      "passwordhistorydays":30,
      "changeoncheckin":"y",
      "changeoncheckout":"y",
      "passwordcharsmin":8,
      "passwordcharsmax":8,
      "passwordalphabeticmin":1,
      "passwordnumericmin":1,
      "passwordalphanumericmin":2,
      "passworduniquemin":1,
      "passworduppercasemin":1,
      "passwordlowercasemin":1,
      "passwordspecialmin":0,
      "passwordspecialmax":0,
      "passwordrepeatedmin":0,
      "passwordrepeatedmax":1,
      "startingchar":"n",
      "isaccountnameallowed":"n",
      "requiredchars":[
         "a",
         "h",
         "j"
      ],
      "allowedchars":[
         "b",
         "t",
         "y",
         "p",
         "u",
         "r",
         "o",
         "k",
         "1",
         "2",
         "=",
         "M",
```

```
         "a",
         "h",
         "j"
      ],
      "disalloweddchars":[
         "7",
         "8",
         "1"
      ]
   }
}
```

All attributes are *optional*, except `policyname`. For attribute definitions refer to Section B.5.3.1, "Retrieve a Password Policy."

B.5.3.4 Get Accounts for Password Policy

Use this API to retrieve a list of accounts for a Password Policy.

> **Note:** You must be an administrator with the *User Manager* Admin Role or the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/passwordpolicy/{policyid}/accounts
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of accounts

Example B–14 Sample JSON Representation of Accounts

```
{
    "accounts":[
        {
            "account":{
                "accountUID":"5bb2c74e1655487c92ecefd5b5270e95",
                "accountName":"dsperson1",
                "targetID":"3ba06e568166493384f86aa5cc7152f1",
                "targetName":"sunds_6.3_target",
                "targetDomain":"needtofix",
                "targetType":"ldap"
                }
        },
        {
            "account":{
               "account":{
                    "accountUID":"c67f93d7a7e44844b24aa43d4cd236e9",
                    "accountName":"person2",
                    "targetID":"75a23e9f30ba456b961a1f5d327e67ef",
                    "targetName":"ldap1_target",
                    "targetDomain":"needtofix",
                    "targetType":"ldap"
                    }
               }
        }
```

```
    ]
}
```

For attribute definitions, refer to Section B.6, "Target Resource" and Section B.7, "Account Resource."

B.5.3.5 Delete a Password Policy

Use this API to delete a Password Policy.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/passwordpolicy/{policyid}
- **Method:** `DELETE`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status 200

B.5.4 Usage Policy Resource

The APIs described in this section include:

- Section B.5.4.1, "Retrieve a Usage Policy"
- Section B.5.4.2, "Update a Usage Policy"
- Section B.5.4.3, "Create a Usage Policy"
- Section B.5.4.4, "Get Grants for Usage Policy"
- Section B.5.4.5, "Delete a Usage Policy"

B.5.4.1 Retrieve a Usage Policy

Use this API to retrieve a Usage Policy.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/usagepolicy/{policyid}
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of Usage Policy

Example B–15 Sample JSON Representation of Usage Policy

```
{
   "usagepolicy":{
      "policyid":"usagepolicy1",
      "policystatus":"active",
      "policyname":"Default Usage Policy",
      "description":"Default Usage Policy",
      "globaldefault":"y",
      "dateorduration":"duration",
      "expireddateminutesfromcheckout":7200,
      "expireddate":"08\/08\/2088",
      "expireddatehour":0,
```

```
"expireddateminutes":0,
"expireddateamorpm":"am",
"timezone":"America\/Los_Angeles",
"usagedates":[
   {
      "day":"saturday",
      "fromhour":"12",
      "frommminutes":"0",
      "fromamorpm":"am",
      "tohour":"12",
      "tominutes":"0",
      "toamorpm":"am"
   },
   {
      "day":"wednesday",
      "fromhour":"12",
      "frommminutes":"0",
      "fromamorpm":"am",
      "tohour":"12",
      "tominutes":"0",
      "toamorpm":"am"
   },
   {
      "day":"sunday",
      "fromhour":"12",
      "frommminutes":"0",
      "fromamorpm":"am",
      "tohour":"12",
      "tominutes":"0",
      "toamorpm":"am"
   },
   {
      "day":"friday",
      "fromhour":"12",
      "frommminutes":"0",
      "fromamorpm":"am",
      "tohour":"12",
      "tominutes":"0",
      "toamorpm":"am"
   },
   {
      "day":"tuesday",
      "fromhour":"12",
      "frommminutes":"0",
      "fromamorpm":"am",
      "tohour":"12",
      "tominutes":"0",
      "toamorpm":"am"
   },
   {
      "day":"thursday",
      "fromhour":"12",
      "frommminutes":"0",
      "fromamorpm":"am",
      "tohour":"12",
      "tominutes":"0",
      "toamorpm":"am"
   },
   {
      "day":"monday",
```

```
            "fromhour":"12",
            "fromminutes":"0",
            "fromamorpm":"am",
            "tohour":"12",
            "tominutes":"0",
            "toamorpm":"am"
         }
         ],
      "allowcheckouttype":"all",
      "scp":{
         "enable":true
      },
      "ssh":{
         "enableInteractive":true,
         "enableNonInteractive":true,
         "enableCommandLogging":true,
         "commandControl":{
            "listType":"whitelist",
            "listValues":[
               "cd",
               "ls"
            ]
         },
         "commandReplacements":[
            {
               "original":"setenv",
               "replaceWith":"set"
            },
            {
               "original":"history",
               "replaceWith":"safehistory"
            }
         ]
      }
   }
}
```

Where:

- **usagepolicy** is a usagepolicy JSON object.
- **policyid** is the Usage Policy's unique identifier.
- **policystatus** is set to active or disabled.
- **policyname** is a name of the policy
- **description** is a description of the policy.
- **globaldefault** indicates whether the policy is the global default policy or not.
- **dateorduration** indicates how the expiration time is calculated.
  - If set to `date`, then `expireddate`, `expireddatehour`, `expireddateminutes`, and `expireddateamorpm` are used.
  - If set to `duration`, then `expireddateminutesfromcheckout` is used.

  Where:
  - **expireddate** is the date of expiration. The format is *MM/dd/yyyy*.
  - **expireddatehour.hour** are integer values between 0 and 12.
  - **expireddateminutes.minutes** are integer values between 0 and 60.

  - **expireddateamorpm** is am or pm.
  - **expireddateminutesfromcheckout** are minutes from checkout.
- **timezone** is a time zone for the Usage Policy.
- **usagedates** is an array, where each value represents the check out time for individual days.
- **day** is a day of the week, where acceptable values are `sunday`, `monday`, `tuesday`, `wednesday`, `thursday`, `friday`, and `saturday`.

Use the following attributes to indicate a range from and to:

- **fromhour** is an integer value between 0 and 12.
- **fromminutes** is a n integer value between 0 and 60.
- **fromamorpm** is am or pm.
- **tohour** is a ***n*** integer value between 0 and 12.
- **tominutes** is a n integer value between 0 and 60.
- **toamorpm** is am or pm.
- **allowcheckoutype** indicates which type of checkout is permitted for the policy.
  - **all**: Choose this option to allow users to check out passwords and sessions.
  - **password** (*default*): Choose this option to allow users to only check out passwords.
  - **session**: Choose this option to allow users to only check out sessions.
- **scp** is the JSON object with attributes specific to SCP.
  - **enable** specifies whether scp is enabled.
- **ssh** is the JSON object with attributes specific to SSH.
- **enableInteractive** specifies whether ssh is enabled for interactive access. It is a boolean whose default is `true`.
- **enableNonInteractive** specifies whether ssh is enabled for non-interactive access. It is a boolean whose default is `true`.
- **enableCommandLogging** specifies whether command logging is enabled. Command logging allows the auditor to view session recordings as an interactive transcript.
- **commandControl** is the JSON Object with attributes specifying the command control constraints.
- **listType** specifies whether the list specified for command control is a whitelist or a blacklist. Only allowed values are considered as `"whitelist"` or `"blacklist"`. If (`""`) appears, the empty string specifies that the list was ignored.
- **listValue** is an array of command regular expressions.
- **commandReplacements** is an array of commands along with their replacements. By default this list is empty.
- **original** is the command name to match with while specifying a replacement for the command.
- **replaceWith** is the command that will replace the original command.

B.5.4.2 Update a Usage Policy

Use this API to update a Usage Policy. You can update all attributes, except `policyid`, and you can update multiple attributes at a time.

> **Note:** You must be an administrator with the *User Manager* Admin Role or the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/usagepolicy/{policyid}
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation of Usage Policy modification
- **Returns on Success:** Status code 200

Example B–16 Sample JSON Representation of Usage Policy Modification

```
{
   "modifications":[
      {
         "modification":{
            "usagedates":[
               {
                  "day":"saturday",
                  "fromhour":"12",
                  "fromminutes":"0",
                  "fromamorpm":"am",
                  "tohour":"12",
                  "tominutes":"0",
                  "toamorpm":"am"
               },
               {
                  "day":"wednesday",
                  "fromhour":"12",
                  "fromminutes":"0",
                  "fromamorpm":"am",
                  "tohour":"12",
                  "tominutes":"0",
                  "toamorpm":"am"
               }
            ]
         }
      },
      {
         "modification":{
            "expireddatehour":2
         }
      },
      {
         "modification":{
            "scp":{
               "enable":false
            }
         }
      },
      {
         "modification":{
```

```
            "ssh":{
               "commandControl":{
                  "listValues":[
                     "cd",
                     "ls.*"
                  ]
               }
            }
         }
      },
      {
         "modification":{
            "ssh":{
               "commandReplacements":[
                  {
                     "original":"setenv",
                     "replaceWith":"set"
                  }
               ]
            }
         }
      }
   ]
}
```

Where:

- **modifications** are an array of modification JSON objects.
- **modification** is a JSON object representing a single attribute.

B.5.4.3 Create a Usage Policy

Use this API to create a Usage Policy.

> **Note:** You must be an administrator with the *User Manager* Admin Role or the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/usagepolicy
- **Method:** `POST`
- **Content-Type:** application/json
- **Body:** JSON representation for Usage Policy creation
- **Returns on Success:** Status code 201
- **SCP:** true/false
- **SSH:** true/false

Example B–17 Sample JSON Representation for Usage Policy Creation

```
{
   "usagepolicy":{
      "policystatus":"active",
      "policyname":"Custom Usage Policy",
      "description":"Custom Usage Policy",
      "globaldefault":"y",
      "dateorduration":"duration",
      "expireddateminutesfromcheckout":7200,
```

```
"expireddate":"08\/08\/2088",
"expireddatehour":0,
"expireddateminutes":0,
"expireddateamorpm":"am",
"timezone":"America\/Los_Angeles",
"usagedates":[
   {
      "day":"saturday",
      "fromhour":"12",
      "fromminutes":"0",
      "fromamorpm":"am",
      "tohour":"12",
      "tominutes":"0",
      "toamorpm":"am"
   },
   {
      "day":"wednesday",
      "fromhour":"12",
      "fromminutes":"0",
      "fromamorpm":"am",
      "tohour":"12",
      "tominutes":"0",
      "toamorpm":"am"
   },
   {
      "day":"sunday",
      "fromhour":"12",
      "fromminutes":"0",
      "fromamorpm":"am",
      "tohour":"12",
      "tominutes":"0",
      "toamorpm":"am"
   },
   {
      "day":"friday",
      "fromhour":"12",
      "fromminutes":"0",
      "fromamorpm":"am",
      "tohour":"12",
      "tominutes":"0",
      "toamorpm":"am"
   },
   {
      "day":"tuesday",
      "fromhour":"12",
      "fromminutes":"0",
      "fromamorpm":"am",
      "tohour":"12",
      "tominutes":"0",
      "toamorpm":"am"
   },
   {
      "day":"thursday",
      "fromhour":"12",
      "fromminutes":"0",
      "fromamorpm":"am",
      "tohour":"12",
      "tominutes":"0",
      "toamorpm":"am"
   },
```

```
        {
           "day":"monday",
           "fromhour":"12",
           "fromminutes":"0",
           "fromamorpm":"am",
           "tohour":"12",
           "tominutes":"0",
           "toamorpm":"am"
        }
        ],
     "allowcheckouttype":"all",
     "scp":{
        "enable":true
     },
     "ssh":{
        "enableInteractive":true,
        "enableNonInteractive":true,
        "enableCommandLogging":true,
        "commandControl":{
           "listType":"whitelist",
           "listValues":[
              "cd",
              "ls"
           ]
        },
        "commandReplacements":[
           {
              "original":"setenv",
              "replaceWith":"set"
           },
           {
              "original":"history",
              "replaceWith":"safehistory"
           }
        ]
     }
  }
}
```


For attribute definitions, refer to Section B.5.4.1, "Retrieve a Usage Policy."

B.5.4.4 Get Grants for Usage Policy

Use this API to retrieve a list of grants for a Usage Policy.

> **Note:** You must be an administrator with the *User Manager* Admin Role or the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/usagepolicy/{policyid}/grantees
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON Representation of grants

Example B–18 Sample JSON Representation of Grants

```
{
    "grantees":[
        {
            "grantee":{
            "accountUID":"16d245784350469cbe25229a7c45af22",
            "accountName":"oidperson10",
            "targetID":"75a23e9f30ba456b961a1f5d327e67ef",
            "targetName":"ldap1_target",
            "targetDomain":"needtofix",
            "targetType":"ldap",
            "grantee":"CrossDomainConnectors",
            "grantType":"role"
            }
        },
        {
            "grantee":{
            "accountUID":"3a7f105a1e45407284cd887f8774700d",
            "accountName":"openLDAPperson2",
            "targetID":"dd9d7a31b39348c79eb23ac46f04d40d",
            "targetName":"openldap_2.3_target",
            "targetDomain":"needtofix",
            "targetType":"ldap",
            "grantee":"opamuser2",
            "grantType":"user"
            }
        }
    ]
}
```

For attribute definitions, refer to Section B.6, "Target Resource" and Section B.7, "Account Resource."

B.5.4.5 Delete a Usage Policy

Use this API to delete a Usage Policy.

> **Note:** You must be an administrator with the *User Manager* Admin Role or the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/usagepolicy/{policyid}
- **Method:** `DELETE`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status 200

B.6 Target Resource

The APIs described in this section include:

- Section B.6.1, "Get Target Attributes"
- Section B.6.2, "Add a Target"
- Section B.6.3, "Verify a Target"

- Section B.6.4, "Retrieve a Target"
- Section B.6.5, "Update a Target"
- Section B.6.6, "Remove a Target"
- Section B.6.7, "Search for Targets"
- Section B.6.8, "Get Available Accounts"
- Section B.6.9, "Retrieve Accounts Registered on a Target"
- Section B.6.10, "Get Target Types"
- Section B.6.11, "Reset Password"
- Section B.6.12, "Show Service Account Password"
- Section B.6.13, "Show Service Account Password (Deprecated)"
- Section B.6.14, "Show Service Account Password History"

B.6.1 Get Target Attributes

Use this API to retrieve a list of the attributes that are associated with all of the target types.

You can use the list of supported target types, along with these attributes, to create the JSON object required to add a target. Refer to Section B.6.2, "Add a Target" for more information.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/attributes
- **Method:** `GET`
- **Content-Type:** NA
- **Returns on Success:** Status code 200 and JSON representation of target types, along with the attributes associated with them.

Sample URI

https://*opam_server_host*:*opam_ssl_port*/opam/target/attributes

Example B–19 JSON Output of Supported Target Types with Attributes

```
{
   "TargetAttributes":[
      {
         "TargetType":"ldap",
         "DisplayName":"ldap",
         "Remote":false,
         "BasicAttributes":[
            {
               "name":"targetName",
               "type":"string",
               "description":"",
               "label":"Name",
               "mask":"false",
               "array":"false",
               "required":"true"
               "readonly":"false"
            },
            {
               "name":"description",
               "type":"string",
```

```
      "description":"",
      "label":"Description",
      "mask":"false",
      "array":"false",
      "required":"false"
   },
   {
      "name":"organization",
      "type":"string",
      "description":"",
      "label":"Organization",
      "mask":"false",
      "array":"false",
      "required":"false"
   },
   {
      "name":"domain",
      "type":"string",
      "description":"",
      "label":"Domain",
      "mask":"false",
      "array":"false",
      "required":"true"
   },
   {
      "name":"host",
      "type":"string",
      "description":"",
      "label":"Host",
      "mask":"false",
      "array":"false",
      "required":"true"
   },
   {
      "name":"port",
      "type":"int",
      "description":"TCP/IP port number used to communicate with the LDAP server.",
      "label":"TCP Port",
      "default":"",
      "mask":"false",
      "array":"false",
      "required":"true"
   },
   {
      "name":"ssl",
      "type":"boolean",
      "description":"Select the check box to connect to the LDAP server using SSL.",
      "label":"SSL",
      "default":"false",
      "mask":"false",
      "array":"false",
      "required":"true"
   },
   {
      "name":"principal",
      "type":"string",
      "description":"The distinguished name with which to authenticate
         to the LDAP server.",
      "label":"Principal",
      "default":"",
```

```
      "mask":"false",
      "array":"false",
      "required":"true"
   },
   {
      "name":"credentials",
      "type":"string",
      "description":"Password for the principal.",
      "label":"Password",
      "default":"",
      "mask":"true",
      "array":"false",
      "required":"true"
   },
   {
      "name":"baseContexts",
      "type":"string",
      "description":"One or more starting points in the LDAP tree that will be used
         when searching the tree. Searches are performed when discovering users from
         the LDAP server or when looking for the groups of which a user is a member.",
      "label":"Base Contexts",
      "default":[

      ],
      "mask":"false",
      "array":"true",
      "required":"true"
   },
   {
      "name":"accountNameAttribute",
      "type":"string",
      "description":"Attribute which holds the account's user name.",
      "label":"Account User Name Attribute",
      "default":"uid",
      "mask":"false",
      "array":"false",
      "required":"true"
   }
],
"AdvancedAttributes":[
   {
      "name":"uidAttribute",
      "type":"string",
      "description":"The name of the LDAP attribute which is mapped
         to the Uid attribute.",
      "label":"Uid Attribute",
      "default":"uid",
      "mask":"false",
      "array":"false",
      "required":"false"
   },
   {
      "name":"accountSearchFilter",
      "type":"string",
      "description":"An optional LDAP filter to control which accounts are returned
         from the LDAP resource. If no filter is specified, only accounts that include
         all specified object classes are returned.",
      "label":"LDAP Filter for Retrieving Accounts",
      "default":"(uid=*)",
      "mask":"false",
```

```
            "array":"false",
            "required":"false"
         },
         {
            "name":"passwordAttribute",
            "type":"string",
            "description":"The name of the LDAP attribute which holds the password.
               When changing an user's password, the new password is set to this attribute.",
            "label":"Password Attribute",
            "default":"userpassword",
            "mask":"false",
            "array":"false",
            "required":"false"
         },
         {
            "name":"accountObjectClasses",
            "type":"string",
            "description":"The object class or classes that will be used when
               creating new user objects in the LDAP tree. When entering more than one
               object class, each entry should be on its own line; do not use commas or
               semi-colons to separate multiple object classes. Some object classes
               may require that you specify all object classes in the class hierarchy.",
            "label":"Account Object Classes",
            "default":[
               "top",
               "person",
               "organizationalPerson",
               "inetOrgPerson"
            ],
            "mask":"false",
            "array":"true",
            "required":"false"
         }
      ]
   }
]
}
```

Where:

- **TargetAttributes** is an array of objects, where each object represents a target type.
- **TargetType** is the target type.
- **DisplayName** is how the target type name should display.
- **BasicAttributes** is an array of objects, where each object represents basic attributes for the target type.
- **AdvancedAttributes** is an array of objects, where each object represents advanced attributes for the target type.
- **name** is the attribute name to use when constructing the target JSON to create a target.
- **type** is the attribute type. Acceptable values include `string`, `int`, `boolean`, or `lov` (list of values).
- **description** is a helpful description of the attribute.
- **label** is how the attribute name should display.
- **default** is a default value for the attribute.

Specify a single value if the `array` parameter is **false** or specify an array of values if `array` is **true**.

- **mask** hides sensitive values, such as credentials.
  - Specify `true` to hide attributes.
  - Specify `false` if hiding attributes is not necessary.
- **array** indicates whether the attribute is single-valued or an array of multiple values.
  - Specify `true` if the attribute is an array of multiple values.
  - Specify `false` if the attribute is single-valued.
- **required** indicates whether the attribute is mandatory or optional.
  - Specify `true` for mandatory attributes.
  - Specify `false` for optional attributes.
- **Remote** indicates whether this target type is supported through a connector server.

B.6.2 Add a Target

Use this API to add a target.

Note:

- You must be an administrator with the *Security Administrator* Admin Role to use this API.
- First, you must obtain a list of attributes for the target type as described in Section B.6.1, "Get Target Attributes." You use these attributes to create the JSON object sent in the body.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target
- **Method:** `POST`
- **Content-Type:** application/json
- **Body:** JSON representation of target for addition/test
- **Returns on Success:** Status code 201 Created and Location

Example B–20 Sample JSON Representation of Target for Addition (ldap TargetType)

```
{
   "target":{
      "targetType":"ldap",
      "targetName":"ldap1-target",
      "host":"opam_server_host",
      "passwordpolicy":"712375b4b7bb453c9482d02535989b53",
      "domain":"berkeley",
      "description":"Ldap target",
      "organization":"ST-US",
      "credentials":"welcome",
      "uidAttribute":"uid",
      "port":"9876",
      "passwordAttribute":"userpassword",
      "principal":"cn=orcladmin",
```

```
      "accountSearchFilter":"(uid=*)",
      "baseContexts":[
         "cn=Users,c=US"
      ],
      "ssl":"false",
      "accountObjectClasses":[
         "top",
         "person",
         "organizationalPerson",
         "inetOrgPerson"
      ],
      "accountNameAttribute":"uid"
   }
}
```

Example B–21 Sample JSON Representation of Target for Addition (lockbox TargetType)

```
{
    "target" : {
        "targetUID" : "62bcfb98f95174ad1900ea2535989b53",
        "targetType" : "targetType",
        "targetName" : "lockbox_target",
        "passwordpolicy" : "passwordpolicy1",
        "passwordchgtime" : "2015-01-12 11:59:39.935",
        "host" : "myhost.us.example.com",
        "domain" : "",
        "description" : "",
        "connectorserverid" : "",
        "targetCustomAttrs":[{"targetCustomAttr" : {
        "attrname" : "attr1"
        "attrvalue" : ["value1"]
         }}]

    }
}
```

Example B–22 Sample JSON Representation of Target for Addition (database TargetType)

```
{
    "target" : {
        "targetType" : "database",
        "targetName" : "db1_target",
        "passwordpolicy" : "712375b4b7bb453c9482d02535989b53",
        "passwordrollover" : "true",
        "host" : "afg1140282",
        "domain" : "adc1140282Domain",
        "description" : "Dbase target for the automation",
        "connectionProperties" : "",
        "dbType" : "Oracle",
        "jdbcUrl" : "jdbc:oracle:thin:@afg1140282.pk.com:11227:db5474",
        "loginPassword" : "password1",
        "loginUser" : "system"
    }
}
```

Example B–23 Sample JSON Representation of Target for Addition (unix TargetType)

```
{
    "target" : {
        "targetType" : "unix",
```

```
        "targetName" : "BackUpUnixTarget",
        "passwordpolicy" : "712375b4b7bb453c9482d02535989b53",
        "passwordrollover" : "true",
        "host" : "myhost.us.example.com",
        "domain" : "US",
        "description" : "Backup system",
        "organization" : "IT",
        "port" : "23",
        "sudoPasswdExpectExpression" : "password",
        "commandTimeout" : "120000",
        "passwordExpectExpressions" :
         "new[\\s](unix[\\s])?password:,new[\\s](unix[\\s])?password([\\s]again)?:",
        "loginShellPrompt" : "$",
        "prePasswdExpectExpression" : "None",
        "sudoAuthorization" : "false",
        "loginUserpassword" : "password1",
        "loginUser" : "aime2"
    }
}
```

Example B–24 Sample JSON Representation of Target for Addition (windows TargetType)

```
{
  "target":{
    "targetType":"windows",
    "targetName":"Windows7Target",
    "connectorserverid":"52d42cf5346f46449a565939dce61d05",
    "passwordpolicy":"9a565939d6f46449a5659352d42cf53",
    "passwordrollover":"false",
    "host":"myhost.us.example.com",
    "domain":"US",
    "description":"Windows7 target system",
    "organization" : "IT",
    "AdminPassword":"password1",
    "AdminName":"MYHOST\Administrator"
  }
}
```

Sample Output

https://*opam_server_host*:*opam_ssl_port*/opam/target /9bbcbbb087174ad1900ea691a2573b61 as the Location.

Where:

- **target** is the target JSON object.
- **targetName** is the name of the target.
- **targetType** is the target type.
- **passwordpolicy** is the Password Policy identifier of the Password Policy applied to the target.
- **passwordrollover** is the flag that indicates whether to enable automatic password recycling for a target's service account.

  If you set this flag to `true`, then Oracle Privileged Account Manager automatically resets the target's service account password based on the settings specified in the Password Policy that applies.

> **Note:** The `passwordrollover` flag is currently not supported for ldap or lockbox targets.

- **connectorserverid** indicates the connector server associated with the target. connectorserverid would be empty, signified by `("")`, for a target using local bundle jars.

All of the other attributes are dynamic and they correspond to the attributes in Section B.6.1, "Get Target Attributes."

B.6.3 Verify a Target

Use this API to verify a target.

> **Note:** First, you must obtain a list of attributes for the target type. Refer to Section B.6.1, "Get Target Attributes," to create the JSON object to be sent in the body.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/test
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation of target for addition/test
- **Returns on Success:** Status code 200

Example B–25 Sample JSON Representation of Target for Addition/Verification

```
{
   "target":{
      "targetType":"ldap",
      "targetName":"ldap1-target",
      "host":"opam_server_host",
      "passwordpolicy":"712375b4b7bb453c9482d02535989b53",
      "domain":"berkeley",
      "description":"Ldap target",
      "organization":"ST-US",
      "credentials":"welcome",
      "uidAttribute":"uid",
      "port":"9876",
      "passwordAttribute":"userpassword",
      "principal":"cn=orcladmin",
      "accountSearchFilter":"(uid=*)",
      "baseContexts":[
         "cn=Users,c=US"
      ],
      "ssl":"false",
      "accountObjectClasses":[
         "top",
         "person",
         "organizationalPerson",
         "inetOrgPerson"
      ],
      "accountNameAttribute":"uid"
   }
}
```

Where:

- **target** is the target JSON object.
- **targetName** is the name of the target.
- **targetType** is the target type.
- **passwordpolicy** is the Password Policy identifier of the Password Policy applied to the target.

All of the other attributes are dynamic and they correspond to the attributes in Section B.6.1, "Get Target Attributes."

B.6.4 Retrieve a Target

Use this API to retrieve a target.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/{targetUID}
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of target

Example B–26 Sample JSON Representation of Target (ldap Target Type)

```
{
   "target":{
      "targetUID":"62bcfb98f95174ad1900ea2535989b53",
      "targetType":"ldap",
      "targetName":"ldap1-target",
      "host":"opam_server_host",
      "domain":"berkeley",
      "description":"Ldap target",
      "organization":"ST-US",
      "credentials":"welcome",
      "uidAttribute":"uid",
      "port":"9876",
      "passwordAttribute":"userpassword",
      "principal":"cn=orcladmin",
      "accountSearchFilter":"(uid=*)",
      "baseContexts":[
         "cn=Users,c=US"
      ],
      "ssl":"false",
      "accountObjectClasses":[
         "top",
         "person",
         "organizationalPerson",
         "inetOrgPerson"
      ],
      "accountNameAttribute":"uid",
   }
}
```

Example B–27 Sample JSON Representation of Target (database Target Type)

```
{
```

```
    "target" : {
        "targetUID" : "62bcfb98f95174ad1900ea2535989b53",
        "targetType" : "database",
        "targetName" : "db1_target",
        "passwordpolicy" : "712375b4b7bb453c9482d02535989b53",
        "passwordrollover" : "true",
        "host" : "afg1140282",
        "domain" : "adc1140282Domain",
        "description" : "Dbase target for the automation",
        "connectionProperties" : "",
        "dbType" : "Oracle",
        "jdbcUrl" : "jdbc:oracle:thin:@afg1140282.us.pk.com:11227:db5474",
        "loginPassword" : "password1",
        "loginUser" : "system"
    }
}
```

Example B–28 Sample JSON Representation of Target (unix Target Type)

```
{
    "target" : {
        "targetUID" : "62bcfb98f95174ad1900ea2535989b53",
        "targetType" : "unix",
        "targetName" : "unix1-target",
        "passwordpolicy" : "712375b4b7bb453c9482d02535989b53",
        "passwordrollover" : "true",
        "host" : "myhost.us.example.com",
        "domain" : "US",
        "description" : "Backup system",
        "organization" : "IT",
        "port" : "23",
        "sudoPasswdExpectExpression" : "password",
        "commandTimeout" : "120000",
        "passwordExpectExpressions" :
        "new[\\s](unix[\\s])?password:,new[\\s](unix[\\s])?password([\\s]again)?:",
        "loginShellPrompt" : "$",
        "prePasswdExpectExpression" : "None",
        "sudoAuthorization" : "false",
        "loginUserpassword" : "password1",
        "loginUser" : "aime2"
    }
}
```

Example B–29 Sample JSON Representation of Target (Windows Target Type)

```
{
  "target":{
    "targetType":"windows",
    "targetName":"Windows7Target",
    "targetAgentKey" :
"wsiaWCKz\/um9kJWTrjz8DaoM5mxnk\/sUIjDyEZrSc4FBHxO8P+3VS39xL8gQs3JuYlS6h+m01N\/5Rg0Y686xCorU=:AQAB"
    "targetUID" : "62bcfb98f95174ad1900ea2535989b53"
    "connectorserverid":"52d42cf53465939dce61d05",
    "passwordpolicy":"9a565659352d42cf53",
    "passwordrollover":"false",
    "host":"myhost.us.example.com",
    "domain":"US",
    "description":"Windows7 target system",
    "organization" : "IT",
```

```
    "AdminPassword":"Password1",
    "AdminName":"SLC05TYZ\Administrator"
  }
}
```

> **Note:** The "targetAgentKey" parameter will display for the agent registered target. It will not display for a normal windows target.

Where:

- **target** is the target JSON object.
- **targetUID** is the target's unique identifier.
- **targetName** is the name of the target.
- **targetType** is target type.
- **passwordrollover** is the flag that indicates whether to enable automatic password recycling for a target's service account.

  If you set this flag to `true`, then Oracle Privileged Account Manager automatically resets the target's service account password based on the settings specified in the Password Policy that applies.

  > **Note:** The `passwordrollover` flag is currently not supported for ldap or lockbox targets.

- **connectorserverid** indicates the connector server associated with the target. connectorserverid would be empty, signified by `("")`, for a target using local bundle jars.

All of the other attributes are dynamic and they correspond to the attributes in Section B.6.1, "Get Target Attributes."

B.6.5 Update a Target

Use this API to update a target.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/{targetUID}
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation of Target Modification
- **Returns on Success:** Status code 200

You can change all of the attributes, except `targetType` and `targetUID`, and you can change multiple attributes at a time.

Example B–30 Sample JSON Object to Modify Target

```
{
```

```
         "modification":{
            "host":"opam_server_host"
         }
      },
      {
         "modification":{
            "port":"6000"
         }
      }
   ]
}
```

Where:

- **targetUID** is the target's unique identifier.
- **modifications** is an array of modification JSON objects.
- **modification** is a JSON object representing the modification of a single attribute.

B.6.6 Remove a Target

Use this API to delete a target.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/{targetUID}
- **Method:** `DELETE`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200

B.6.7 Search for Targets

Use this API to search for a target using any of the following request parameters:

- `type`
- `name`
- `hostname`
- `domain`
- `description`
- `org`
- `customattrname`
- `customattrvalue`

All of these parameters are *optional*.

Note: ■You must be an administrator with the *User Manager* Admin Role, *Security Administrator* Admin Role, or *Security Auditor* Admin Role to use this API.

- There should be one customattrvalue per customattrname.

  For example:

  ```
  https://opam_server_host:opam_ssl_port/opam/target/search
  ?customattrname=location&customattrvalue=US&customattrnam
  e=owner&customattrvalue=john
  ```

  The preceding example will search all targets that have custom attribute pairs, which have US as location and john as owner.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/search?param1=value1&param2=value2
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of Target Collection

Sample URIs:

`https://opam_server_host:opam_ssl_port/opam/target/search?` : Returns all targets

`https://opam_server_host:opam_ssl_port/opam/target/search?type=ldap&org=us` : Returns all targets whose type contains ldap and org contains us.

Example B–31 Sample JSON Representation of Target Collection

```
{
   "Target Collection":[
      {
         "target":{
            "uri":"https:\/\/opam_server_host:opam_ssl_port\/opam\/target\
               /9bbcbbb087174ad1900ea691a2573b61",
            "type":"ldap",
            "name":"person1-ldap",
            "host":"opam_server_host",
            "domain":"berkeley"
            "description" : "Ldap target"
         }
      },
      {
         "target":{
            "uri":"https:\/\/opam_server_host:opam_ssl_port\/opam\/target\
               /ac246a162ce948c7b1cdcc17dfc92c15",
            "type":"ldap",
            "name":"person1-ldap2",
            "host":"opam_server_host:opam_ssl_port",
            "domain":"berkeley"
            "description" : "Ldap target"
         }
      }
```

```
   ]
}
```

Where:

- **Target Collection** is an array of target JSON objects.
- **target** is the target JSON object.
- **uri** is the target resource URI.
- **type** is the target type.
- **hostname** is the target's host name.
- **name** is the target name.
- **org** is the target's organization.
- **domain** is the target's domain.
- **description** is a description of the target system.

B.6.8 Get Available Accounts

Use this API to retrieve all of the accounts present on the target system.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/{targetUID}/availableaccounts
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 OK and JSON representation of account collection

Example B–32 Sample JSON Representation of Account Collection

```
{
   "AvailableAccounts":[
      {
         "accountName":"SCOTT",
         "accountUid":"SCOTT"
      },
      {
         "accountName":"BLAKE",
         "accountUid":"BLAKE "
      },
      {
         "accountName":"JONES",
         "accountUid":"JONES"
      }
   ]
}
```

Where:

- **AvailableAccounts** is an array of the accounts present on the target system.
- **accountName** is the account name.
- **accountUID** is the account's unique identifier.

B.6.9 Retrieve Accounts Registered on a Target

Use this API to retrieve all the accounts on the target that are registered with Oracle Privileged Account Manager.

> **Note:** You must be an administrator with the *User Manager* Admin Role, *Security Administrator* Admin Role, or *Security Auditor* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/{targetUID}/accounts
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of URI collection of accounts Server

Example B–33 Sample JSON Representation of URI Collection of Accounts

```
{
   "URI Collection":[
      {
         "account":{
            "uri":"https:\/\/opam_server_host:opam_ssl_port\/opam\/account\
               /3740553e999a4f6aa8e8f9286d320cb4",
            "accountName":"sherlock"
         }
      },
      {
         "account":{
            "uri":"https:\/\/opam_server_host:opam_ssl_port\/opam\/account\
               /c11066278022489aad758aec69d9727d",
            "accountName":"root"
         }
      }
   ]
}
```

Where:

- **URI Collection** is an array of accounts on a target that are registered with Oracle Privileged Account Manager.
- **account** is the account JSON object.
- **uri** is the account's URI.
- **accountName** is the account name.

B.6.10 Get Target Types

Use this API to retrieve a list of all supported target types.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/types
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of supported target types

Example B–34 Sample JSON Representation of Supported Target Types

```
{
   "targettypes":[
      "ldap",
      "unix",
      "database",
      "lockbox"
      "sapum"
      "sapume"
      "unix"
      "windows"
   ]
}
```

Where: "**targettypes**" are the supported target types.

B.6.11 Reset Password

Use this API to reset the password on the target's service account.

Note:

- You must be an administrator with the *Security Administrator* Admin Role to use this API.
- Refer to Chapter 7, "Working with Service Accounts" for information about service accounts.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/{targetUID}/resetpassword
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200

Example B–35 Sample JSON Representation of the New Password

```
{
   "password":"password1"
}
```

or

```
{
   "autogen":"true"
}
```

Where:

- **targetUID** is the target's unique identifier.
- **password** is the password to assign to the service account.
- **autogen** is the flag that controls whether to automatically generate the password or not. The default value of this flag is `false`.

B.6.12 Show Service Account Password

Use this API to retrieve and display the service account password.

Note:

- You must be an administrator with the *Security Administrator* Admin Role to use this API.
- Refer to Chapter 7, "Working with Service Accounts" for information about service accounts.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/{targetUID}/showpassword
- **Method:** `GET`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of service account

Example B–36 Sample JSON Representation of Account Token

```
{
  "serviceAccount" : {
    "targetName" : "APILDAP",
    "targetUID" : "62bcfb98f95174ad1900ea2535989b53",
    "targetAccount" : "cn=admin",
    "targetPassword" : "password1",
    "targetPasswordChangeTime" : " 2013-01-27 02:58:13.259"
  }
}
```

Where:

- **targetUID** is the target's unique identifier.
- **targetName** is the name of the target.
- **targetAccount** is the service account on the target.
- **targetPassword** is the service account password.
- **targetPasswordChangeTime** is the time when the password was modified.

B.6.13 Show Service Account Password (*Deprecated*)

> **Note:** This API has been deprecated. Oracle recommends that you use the Show Service Account Password API in Section B.6.12, "Show Service Account Password."

Use this API to retrieve and display the service account password.

> **Note:**
>
> - You must be an administrator with the *Security Administrator* Admin Role to use this API.
> - Refer to Chapter 7, "Working with Service Accounts" for information about service accounts.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/{targetUID}/showpassword
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of service account

Example B–37 Sample JSON Representation of Account Token

```
{
  "serviceAccount" : {
    "targetName" : "APILDAP",
    "targetUID" : "62bcfb98f95174ad1900ea2535989b53",
    "targetAccount" : "cn=admin",
    "targetPassword" : "password1",
    "targetPasswordChangeTime" : " 2013-01-27 02:58:13.259"
  }
}
```

Where:

- **targetUID** is the target's unique identifier.
- **targetName** is the name of the target.
- **targetAccount** is the service account on the target.
- **targetPassword** is the service account password.
- **targetPasswordChangeTime** is the time when the password was modified.

B.6.14 Show Service Account Password History

Use this API to retrieve and display the service account password history.

Note:

- You must be an administrator with the *Security Administrator* Admin Role to use this API.
- Refer to Chapter 7, "Working with Service Accounts" for information about service accounts.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/target/{targetUID}/showpasswordhistory
- **Method:** `GET`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of service account Server

Example B–38 Sample JSON Representation of Target Token

```
{
    "targetToken": {
        "targetName": "SessionMgr_Target",
        "targetUID": "62bcfb98f95174ad1900ea2535989b53",
        "passwordHistory": [
            {
                "targetPassword": "password1",
                "modificationTime": "1383078344"
            },
            {
                "targetPassword": "4PkVerh7",
                "modificationTime": "1383078329"
            },
            {
                "targetPassword": "l9yAigqj",
                "modificationTime": "1383078314"
            },
            {
                "targetPassword": "password1",
                "modificationTime": "1383010874"
            }
        ]
    }
}
```

Where:

- **targetUID** is the target's unique identifier.
- **targetName** is the name of the target.
- **passwordHistory** is the service account password history.
- **targetPassword** is the service account password.
- **modificationTime** (UTC time in seconds) is the time when the password was modified.

Password history results are sorted by modification time, where the most recent results will be at the top.

B.7 Account Resource

The APIs described in this section include:

B.7.1 Add an Account to a Target

Use this API to add an account to the target. This API does not create an account on the target system, but it registers the existing account with the Oracle Privileged Account Manager target.

Note:

- You must never use the same account as the service account *and* as a privileged account to be managed by Oracle Privileged Account Manager.
- You must be an administrator with the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/accoun
- **Method:** `POST`
- **Content-Type:** application/json
- **Body:** JSON representation for account addition/verification
- **Returns on Success:** Status code 201 and Location

Example B–39 Sample JSON Representation of Account for Addition/Verification

```
{
   "account":{
      "accountName":"admin",
      "description" : "maintenance account on the machine",
      "password" : "password1",
      "passwordpolicy":"passwordpolicy2",
      "shared":"true",
      "targetUID":"62bcfb98f95174ad1900ea2535989b53"
      "accountCustomAttrs": [{"accountCustomAttr": {
        "attrname": "attr1",
        "attrvalue": ["100"]
          }}]

   }
}
```

Where:

- **account** is the account JSON object.
- **accountName** is the name of the account.
- **description** is a description of the account. This attribute is *optional*.
- **password** is the account password. This attribute is *optional*.
- **passwordpolicy** is the policy ID of the Password Policy applicable to the account. This parameter is *optional*. By default, this parameters uses the global Default Password Policy.
- **shared** indicates the shared status of the account. This value is a Boolean and the default setting is `false`.
- **targetUID** is the target's unique identifier.

B.7.2 Get Applicable Usage Policy for the Account

Use this API to get the applicable Usage Policy for an account.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/*accountUID*/usagepolicy
- **Method:** `GET`
- **Content-Type:** NA
- **Returns on Success:** Status code 200 and JSON representation of the Usage Policy

Example B–40 Sample JSON Representation of the Usage Policy

```
{"usagepolicy":
    {
        "policyid":"bafd53072bbb442db185dca18bd00e69",
        "policyname":"usage_policy_anytime"
    }
}
```

Where:

- **usagepolicy** is the Usage Policy JSON object.
- **policyid** is the Usage Policy's unique identifier.
- **policyname** is a name of the policy

B.7.3 Grant a User/Role Access to an Account

Use this API to grant a user or role access to an account. Multiple users and roles can be granted the access at a time.

> **Note:** You must be an administrator with the *User Manager* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation for adding grantees
- **Returns on Success:** Status code 200

Example B–41 Sample JSON Representation for Adding Grantees

```
{
   "modifications":[
      {
         "modification":{
            "usagepolicy":"712375b4b7bb453c9482d02535989b53",
            "role":"opamgroup1",
            "operation":"add"
         }
      },
      {
         "modification":{
            "usagepolicy":"usagepolicy1",
            "user":"opamuser1",
            "operation":"add"
         }
      }
```

```
   ]
}
```

Where:

- **accountUID** is the account's unique identifier.
- **modifications** are an array of modification JSON objects.
- **modification** is a JSON object representing the modification of a single attribute.
- **role** indicates that a group has to be granted an access. This parameter value is the group name.
- **user** indicates that a user has to be granted an access. This parameter value is the user login id.
- **usagepolicy** indicates the Usage Policy identifier to be applied to the grant.
- **operation** indicates the type of operation to be performed. Acceptable values include:
  - **add** indicates grant.
  - **delete** indicates revocation.
  - **replace** indicates replacement of usagepolicy with a new value.

B.7.4 Add or Remove a CSF Map-Key for an Account

Use this API to add a CSF map-key to an account or remove the map-key from an account. You can add or remove multiple map-keys at a time.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to use this API.

- **URI**: https:///*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation for adding keymaps
- **Returns on Success:** Status code 200

Example B–42 Sample JSON Representation for Map-Keys Addition/Removal

```
{
    "modifications": [
        {
            "modification": {
                "keymap":
"[app1][sd45kjlf4g][t3://myhost:2001][weblogic][password]",
                "operation": "add"
            }
        },
        {
            "modification": {
                "keymap": "[hrmap][hrkey2][t3://myhost:2001][weblogic][password]",
                "operation": "delete"
            }
        }
```

```
    ]
}
```

Where:

- **accountUID** is the account's unique identifier.
- **modifications** is an array of modification JSON objects.
- **modification** is a JSON object representing the modification of a single attribute.
- **keymap** is the map-key to be added or removed. The map-key must be in the following format:

  ```
  [csfmap][csfkey][Administration Server Url][username][password]
  ```

- **operation** indicates the type of operation to be performed. Acceptable values include:
  - **add** indicates addition of map-key.
  - **delete** indicates removal of map-key.

B.7.5 Search Accounts

Use this API to search accounts using one or more of the following search request parameters:

- `type`
- `domain`
- `description`
- `name`
- `accountname`
- `customattrname`
- `customattrvalue`

All of these parameters are *optional*.

Note:
- You must be an administrator with the *User Manager* Admin Role, the *Security Auditor* Admin Role, or the *Security Administrator* Admin Role to use this API.
- There should be one customattrvalue per customattrname.

  For example:

  ```
  https://opam_server_host:opam_ssl_port/opam/account/search?customattrname=location&customattrvalue=US&customattrname=owner&customattrvalue=john
  ```

  The preceding example will search all targets that have custom attribute pairs, which have US as location and john as owner.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/search?
- **Method:** `GET`
- **Content-Type:** NA

- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of account collection

Example B–43 Sample JSON Representation of Account Collection

```
{
    "AccountCollection" : [
        {
            "account" : {
            "shared" : false,
            "passwordchangetime" : 1383072107,
            "targetUID" : "62bcfb98f95174ad1900ea2535989b53",
            "domain" : "needtofix",
            "targetName" : "sunds_6.3_target",
            "targetType" : "ldap",
            "accountlevelstatus" : "checkedIn",
            "description" : "",
            "accountName" : "dsperson1",
            "uri" : "https://localhost:7002/opam/account/35e2709edf0443edae8f67727d937bec",
            "accountUID" : "35e2709edf0443edae8f67727d937bec"
        }
            },
        {
            "account" : {
            "shared" : false,
            "passwordchangetime" : 1383072107,
            "targetUID" : "62bcfb98f95174ad1900ea2535989b53",
            "domain" : "needtofix",
            "targetName" : "sunds_6.3_target",
            "targetType" : "ldap",
            "accountlevelstatus" : "checkedIn",
            "description" : "",
            "accountName" : "dsperson10",
            "uri" : "https://localhost:7002/opam/account/0a1ee2cb17e345cdb537a2f05e11e93c",
            "accountUID" : "0a1ee2cb17e345cdb537a2f05e11e93c"
            }
        }

    ],
  "count" : 2
}
```

Where:

- **account** is the account JSON object.
- **shared** indicates the shared status of the account. This value is a Boolean and the default setting is `false`.
- **accountlevelstatus** indicates whether the account has been checked in by anyone. Acceptable values are `checkedIn` and `checkedOut`.
- **description** is a description of the account. This attribute is *optional*.
- **accountName** is the name of the account.
- **accountUID** is the account's unique identifier.
- **passwordchangetime** is the time when the password was modified.

For all other attribute definitions, refer to Section B.6, "Target Resource."

B.7.6 Search Assigned Accounts

Use this API to search assigned accounts using one or more of the following search request parameters:

- `type`
- `domain`
- `description`
- `name`
- `accountname`

All of these parameters are *optional*.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/myaccounts/search?
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of account collection

Example B–44 Sample JSON Representation of Account Collection

```
{
  "AccountCollection": [
    {
      "account": {
        "uri": "https://myhost:7002/opam/account/aa243a9323974eca84d4141193ca58e1",
        "accountUID": "aa243a9323974eca84d4141193ca58e1",
        "accountName": "account1",
        "description": ""8759",
        "targetUID": "62bcfb98f95174ad1900ea2535989b53",
        "targetName": "kiki",
        "targetType": "lockbox",
        "domain": ""
        "host": "kiki"
      }
    }
  ],
{
  "AccountCollection": [
    {
      "account": {
        "uri": "https://myhost:7002/opam/account/086931f6816647f0a4c0ca6b28055739",
        "accountUID": "086931f6816647f0a4c0ca6b28055739",
        "accountName": "hello",
        "description": "8759",
        "targetUID": "62bcfb98f95174ad1900ea2535989b53",
        "targetName": "lockbox2",
        "targetType": "lockbox",
        "domain": ""
        "host": "myhost.us.example.com"
      }
    }
  ],
    "count": 2
}
```

Where:

- **account** is the account JSON object.
- **accountUID** is the account's unique identifier.
- **accountName** is the name of the account.
- **description** is a description of the account. This attribute is *optional*.

For all other attribute definitions, refer to Section B.6, "Target Resource."

B.7.7 Retrieve an Account

Use this API to retrieve an account.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of account Server

Example B–45 Sample JSON Representation of Account

```
{
   "account":{
      "accountUID":"aa243a9323974eca84d4141193ca58e1",
      "description":"8759",
      "targetUID":"62bcfb98f95174ad1900ea2535989b53",
      "accountName":"account1",
      "shared":false,
      "keymaps":[],
      "passwordpolicy":"passwordpolicy1",
      "accountlevelstatus":"checkedIn",
      "passwordchangetime":"1421107647",
   }
}
```

Where:

- **account** is the account JSON object.
- **accountUID** is the account's unique identifier.
- **accountName** is the name of the account.
- **passwordpolicy** is the policy ID of the Password Policy applicable to the account.
- **shared** indicates the shared status of the account. This value is a Boolean and the default setting is `false`.
- **targetUID** is target's unique identifier.
- **accountlevelstatus** indicates whether the account has been checked in by anyone. Acceptable values are `checkedIn` and `checkedOut`.
- **protocol** is the protocol used to connect to the Oracle Privileged Session Manager server.
- **port** is the port used to connect to the Oracle Privileged Session Manager server.

B.7.8 Retrieve Grantees on an Account

Use this API to retrieve all the grantees of an account. A grantee can be a user or a role.

> **Note:** You must be an administrator with the *User Manager* Admin Role or the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}/grantees
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of Grantees

Example B–46 Sample JSON Representation of Grantees

```
{
   "grantees":{
      "users":[
         "opamuser1"
      ],
      "roles":[
         "opamgroup1"
      ]
   }
}
```

Where:

- **grantees** are grantees of the account.
- **user**s are the users who have been granted the account. Each value is the user's login ID/UID.
- **roles** are the groups or roles who have been granted the account. Each value is a group name.

B.7.9 Retrieve Users Who Checked Out an Account

Use this API to retrieve a list of all users who have currently checked out an account.

> **Note:** You must be an administrator with the *User Manager* Admin Role, the *Security Auditor* Admin Role, or the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}/whocheckedout
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA

- **Returns on Success:** Status code 200 and JSON representation of users who checked out the account.

Example B–47 Sample JSON Representation of Users Who Checked Out the Account

```
{
  "users": [
    {
        "user": {
        "uid": "user_manager",
        "expiryTime": "1382147587",
        "checkoutTime": "1381715587",
        "checkoutUID": "f499b76719ba4d0aa30487e58316def3",
        "checkoutType": "password",
        "transcriptURL": ""
  }
    },
      {
        "user": {
        "uid": "user_manager",
        "expiryTime": "1382147587",
        "checkoutTime": "1381715587",
        "checkoutUID": "f499b76719ba4d0aa30487e58316def3",
        "checkoutType": "session",
        "transcriptURL": "https://myhost:2001/opam/checkout/dee8383184664ddfa09f454d0a9a023d/
          transcript"
      }
    }
  ]
}
```

Where:

- **transcriptURL** is the URL you use to access the session transcript.
- **checkoutType** indicates whether the checkout was a session checkout or a password checkout.
- **checkoutUID** is the unique ID for the checkout.

B.7.10 Check Out an Account

Use this API to check out an account.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/v1/{accountUID}/checkout
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of account token

Example B–48 Sample JSON Representation of Account Token

```
{
   "accountToken":{
      "accountName":"admin",
      "accountUID":"3f74a85e39e64432ba917a2e60fa15aa",
      "accountPassword":"GJN8p2ol"
```

```
   }
}
```

Where:

- **accountUID** is the account's unique identifier.
- **accountName** is the name of the account.
- **accountpassword** is the account password.

> **Note:** In version v1, upon a repeat checkout, account token is returned along with a message in that account is already checked out. In earlier versions, only an error message was sent. This behavior is retained in (https://opam_server_host:opam_ssl_port/opam/account/{account UID}/checkout).

B.7.11 Get All Checked Out Accounts

Use this API to retrieve a list of all accounts that have been checked out by the logged in user.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/mycheckouts
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of account collection

Example B–49 Sample JSON Representation of Account Collection

```
{
  "Checkouts": [
    {
      "uri": "https://myhost:7002/opam/account/b0e7ae053afb45658da4e3a0453bffec",
      "accountUID": "b0e7ae053afb45658da4e3a0453bffec",
      "accountName": "dduck",
      "status": "checkedOut",
      "targetUID": "62bcfb98f95174ad1900ea2535989b53",
      "targetName": "unix1-target",
      "targetType": "unix",
      "domain": "US",
      "expiryTime": "1371945854",
      "checkoutUID": "b97b2de6a80b40c48f873067027ac476",
      "checkoutType": "session",
      "transcriptURL": "https://myhost:2001/opam/account/checkout/b97b2de6a80b40c48f873067027ac476/
           transcript"
    },
    {
      "uri": "https://myhost:7002/opam/account/b0e7ae053afb45658da4e3a0453bffec",
      "accountUID": "b0e7ae053afb45658da4e3a0453bffec",
      "accountName": "dduck",
      "status": "checkedOut",
      "targetUID": "62bcfb98f95174ad1900ea2535989b53",
      "targetName": "unix1-target",
      "targetType": "unix",
```

```
      "domain": "US",
      "expiryTime": "1371940624",
      "checkoutUID": "bf43672ffd3a43018cdfde9b78bf1691",
      "checkoutType": "password",
      "transcriptURL": ""
    }
  ]
}
```

Where:

- **accountUID** is the account's unique identifier.
- **accountName** is the name of the account.
- **checkoutUID** is the unique ID for the checkout.
- **checkoutType** indicates whether the checkout was a session checkout or a password checkout.
- **transcriptURL** is the URL to access the session transcript.

For all other attribute definitions, refer to Section B.6, "Target Resource."

B.7.12 Get Session Checkout Instructions

Use this API to get information to help you perform a session checkout.

> **Note:** For more information about password and session checkouts, refer to Section 9.5, "Checking Out Privileged Accounts" and Section 9.5.3, "Checking Out Privileged Account Sessions."

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}/checkout/session/instructions
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of output

Example B–50 Sample JSON Representation of Session Checkout Instructions

```
{
  "sessionCheckoutInstructions": {
    "accountName": "dduck",
    "targetName": "bkottaha-unix",
    "port": 1222,
    "instruction": "ssh -p <port> <opamuser>:<targetname>:<accountname>@
       <sessionmgrhost>\n Use opam password on password prompt"
  }
}
```

Where:

- **accountName** is the name of the account.
- **targetName** is the name of the target.
- **port** is the port that Session Manager listens to for connections.

- **instruction** is the information required to perform a session checkout.

B.7.13 Checkout History for an Account

Use this API to search for an account's checkout history using one or more of the following parameters:

- `from`: Specify start time in seconds (UTC) (*required*).
- `to`: Specify end time in seconds (UTC) (*required*).
- `uid`: Specify the userID (*optional*).
- `pattern`: Specify the command that was executed or a term in the log (*optional*).
- `size`: Specify the number of array elements to be returned (*optional*).

Use the `from` and `to` parameters to specify the time period in which the checkouts were running.

> **Note:** You must be an administrator with the *User Manager* or *Security Administrator* Admin Role to access this query.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}/checkouts/historical/search?param1=val1
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of output

Sample URL Output

https://*myhost*:7002/opam/account/8d9e9ce750da4aedac3ffbea0d28a73a/checkouts/historical/search?from=123&to=1372893007&size=2&pattern=ls

Example B–51 Sample JSON Representation of Account Checkout History

```
{
   "checkouts":[
      {
         "checkout":{
            "accountName":"itsupport",
            "targetName":"unixTarget",
            "uid":"end_user",
            "starttime":"1404691650",
            "endtime":"1404691654",
            "recordingType":"text\/plain",

"transcriptURL":"https:\/\/myhost:2001\/opam\/checkout\/c3bcb3366581420d9d8166810c1c72da\/transcript",
            "transcript":"\/checkout\/c3bcb3366581420d9d8166810c1c72da\/transcript"
         }
      },
      {
         "checkout":{
            "accountName":"itsupport",
            "targetName":"unixTarget",
```

```
            "uid":"end_user",
            "starttime":"1404691378",
            "endtime":"1404691387",
            "recordingType":"text\/html",

"transcriptURL":"https:\/\/myhost:2001\/opam\/checkout\/b869b1d8a48a4b459adaff010c887543\/transcript",
            "metadata":"\/checkout\/b869b1d8a48a4b459adaff010c887543\/metadata",
            "transcript":"\/checkout\/b869b1d8a48a4b459adaff010c887543\/transcript"
         }
      }
   ],
   "totalcount":5,
   "returncount":5
}
```

Where:

- **transcriptURL** is the URL you use to access the session transcript.
- **checkoutType** indicates whether the checkout was a session checkout or a password checkout.
- **checkoutUID** is the unique ID for the checkout.
- **totalcount** is the number of actual search results.
- **returncount** is the number of search results that were actually returned. This is determined by size.
- **recordingType** is available in the plain text (`"text/plain"`) or interactive (`"text/html"`) formats.
- **metadata** is the relative link to base opam url for the xml metadata for the session.

> **Note:** The metadata attribute is absent if there is no metadata.

- **video** is the relative link to the video
- **transcript** is the relative link for the transcript. This transcript can be in the plain text or html formats.

For all other attribute definitions, refer to Section B.7, "Account Resource."

B.7.14 Checkout History

Use this API to search for the checkout history of all accounts, using one or more of the following parameters:

- `from`: Specify start time in seconds (UTC) (*required*).
- `to`: Specify end time in seconds (UTC) (*required*).
- `targetname`: Specify the name of a target on which to search (*optional*).
- `accountname`: Specify the name of an account to search (*optional*).
- `uid`: Specify the userID (*optional*).
- `pattern`: Specify the command that was executed or a term in the log (*optional*).
- `size`: Specify the number of array elements to be returned (*optional*).

Use the `from` and `to` parameters to specify the time period in which the checkouts were running.

> **Note:** You must be an administrator with the *Security Auditor* Admin Role to access this query.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/checkout/historical/search?param1=val1
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of output

Sample URL

https://*myhost*:7002/opam/checkout/historical/search?from=123&to=1472816146&size=2&pattern=ls&accountname=a&targetname=h&uid=u

Example B–52 Sample JSON Representation of Checkout History

```
{
   "checkouts":[
      {
         "checkout":{
            "accountName":"itsupport",
            "targetName":"unixTarget",
            "uid":"end_user",
            "starttime":"1404691650",
            "endtime":"1404691654",
            "recordingType":"text\/plain",


"transcriptURL":"https:\/\/myhost:2001\/opam\/checkout\/b869b1d8a48a4b459adaff010c
887543\/transcript",

           "transcript":"\/checkout\/c3bcb3366581420d9d8166810c1c72da\/transcript
         }
      },
      {
         "checkout":{
            "accountName":"itsupport",
            "targetName":"unixTarget",
            "uid":"end_user",
            "starttime":"1404691378",
            "endtime":"1404691387",
            "recordingType":"text\/html",


"transcriptURL":"https:\/\/myhost:2001\/opam\/checkout\/b869b1d8a48a4b459adaff010c
887543\/transcript",

            "metadata":"\/checkout\/b869b1d8a48a4b459adaff010c887543\/metadata",
           "transcript":"\/checkout\/b869b1d8a48a4b459adaff010c887543\/transcript"
         }
      },
```

```
        {
           "checkout":{
              "accountName":"SystemAdmin",
              "targetName":"WinTarget",
              "uid":"end_user",
              "starttime":"1403501578",
              "endtime":"1403501593",
              "recordingType":"video",
              "video":"\/checkout\/bde06872949740a59dc5a702d8aca48e\/video",
              "metadata":"\/checkout\/bde06872949740a59dc5a702d8aca48e\/metadata"
           }
        }

     ],
     "totalcount":5,
     "returncount":5
}
```


Where:

- **transcriptURL** is the URL you use to access the session transcript.
- **checkoutType** indicates whether the checkout was a session checkout or a password checkout.
- **checkoutUID** is the unique ID for the checkout.
- **totalcount** is the number of actual search results.
- **returncount** is the number of search results that were actually returned. This is determined by size.
- **recordingType** is available in the plain text (`"text/plain"`) or interactive (`"text/html"`) formats.
- **metadata** is the relative link to base opam url for the xml metadata for the session.

> **Note:** The metadata attribute is absent if there is no metadata.

- **video** is the relative link to the video
- **transcript** is the relative link for the transcript. This transcript can be in the plain text or html formats.

For all other attribute definitions, refer to Section B.7, "Account Resource."

B.7.15 Check In an Account

Use this API to check in an account.

A checkout can be a *password* checkout or *session* checkout. You can individually check in each checkout by using its `checkoutUID` or you can check in all of the checkouts for an account. In this publication, the term "account checkout" generally refers to the latter case.

> **Note:** To do a force-check in, you must be an administrator with the *User Manager* Admin Role.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}/checkin
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200

Sample JSON Representations of Account Check Ins

The following examples illustrate different types of Force Check Ins

- Example B–53, "Self Check In a Password or Session Checkout"
- Example B–54, "Force Account Check In (Both Password and Session) for All Users"
- Example B–55, "Force Account Check In (Both Password and Session) for a Single User"
- Example B–56, "Force Check In a Password or Session"

Example B–53 Self Check In a Password or Session Checkout

```
{
  "checkoutUID":"9c3c5d687d414a57b7dbda0692c9b06d"
}
```

Example B–54 Force Account Check In (Both Password and Session) for All Users

```
{
  "force":"true"

}
```

Example B–55 Force Account Check In (Both Password and Session) for a Single User

```
{
  "force" : "true",
  "userid" : "person1"
}
```

Example B–56 Force Check In a Password or Session

```
{
  "force" : "true",
  "checkoutUID" : "9c3c5d687d414a57b7dbda0692c9b06d",
}
```

Note: If you want to perform an account check in (for both password or session), you do not have to provide any content in the JSON body.

Where:

- **force** is a flag that indicates a force check-in. The default value of this flag is `false`.
- **userid** is the user who is to be force-checked in. The default action is to force-check in all users that have checked out the account.
- **checkoutUID** is the unique identifier for a checkout.

B.7.16 Verify an Account

Use this API to verify whether the account is present on the target system.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/test
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation for account addition/verification
- **Returns on Success:** Status code 200

Example B–57 Sample JSON Representation of Account Addition/Verification

```
{
   "account":{
      "accountName":"admin",
      "description" : "maintenance account on the machine"
      "password" : "password1"
      "passwordpolicy":"passwordpolicy2",
      "shared":"true",
      "targetUID":"62bcfb98f95174ad1900ea2535989b53"
   }
}
```

Where:

- **account** is the account JSON object.
- **accountName** is the name of the account.
- **description** is a description of the account. This attribute is *optional*.
- **password** is the account password. This attribute is *optional*.
- **passwordpolicy** is the policy ID of the Password Policy applicable to the account. This parameter is *optional*. By default, this parameters uses the global Default Password Policy.
- **shared** indicates the shared status of the account. This value is a Boolean and the default setting is `false`.
- **targetUID** is the target's unique identifier.

B.7.17 Update an Account

Use this API to update an account. You can change multiple attributes at a time. Only passwordpolicy, description, and shared attributes can be updated.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation of account modifications
- **Returns on Success:** Status code 200

Example B–58 Sample JSON Representation of Account Modifications

```
{
   "modifications":[
      {
         "modification":{
            "passwordpolicy":"passwordpolicy2"
         }
      },
      {
         "modification":{
            "shared":"false"
         }
      }
   }
}
```

Where:

- **accountUID** is the account's unique identifier.
- **modifications** are an array of modification JSON objects.
- **modification** is a JSON object representing the modification of a single attribute.

B.7.18 Remove an Account

Use this API to remove an account.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID
- **Method:** `DELETE`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200

Where:

- **accountUID** is the account's unique identifier.

B.7.19 Remove a User's/Role's Access to an Account

Use this API to remove a user's access or a role's access to an account. You can revoke multiple user and role grants at a time.

Note: You must be an administrator with the *User Manager* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation for removing grantees
- **Returns on Success:** Status code 200

Example B–59 Sample JSON Representation for Removing Grantees

```
{
   "modifications":[
      {
         "modification":{
            "usagepolicy":"usagepolicy1",
            "role":"opamgroup1",
            "operation":"delete"
         }
      },
      {
         "modification":{
            "usagepolicy":"usagepolicy1",
            "user":"opamuser1",
            "operation":"delete"
         }
      }
   ]
}
```

Where:

- **accountUID** is the account's unique identifier.
- **modifications** are an array of modification JSON objects.
- **modification** is a JSON object representing a single modification.
- **role** indicates that a group has to be granted an access. This parameter value is the group name.
- **user** indicates that a user has to be granted an access. This parameter value is the user login id.
- **usagepolicy** indicates the Usage Policy identifier to be applied to the grant.
- **operation** indicates the type of operation to be performed. Acceptable values include:
  - **add** indicates a grant.
  - **delete** indicates a revocation.
  - **replace** indicates the replacement of the usagepolicy with a new value.

B.7.20 Show Password

Use this API to retrieve and display the password associated with an account.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role or you must have checked out the account to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}/showpassword
- **Method:** `GET`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of account token

Example B–60 Sample JSON Representation of Account Token

```
{
   "accountToken":{
      "accountName":"admin",
      "accountUID":"3f74a85e39e64432ba917a2e60fa15aa",
      "accountPassword":"GJN8p2ol"
   }
}
```

Where:

- **accountUID** is the account's unique identifier.
- **accountName** is the name of the account.
- **accountPassword** is the account password.

B.7.21 Show Password (*Deprecated*)

> **Note:** This API has been deprecated. Oracle recommends that you use the Show Password API in Section B.7.20, "Show Password."

Use this API to retrieve and display the password associated with an account.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role or you must have checked out the account to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}/showpassword
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of account token

Example B–61 Sample JSON Representation of Account Token

```
{
```

```
   "accountToken":{
      "accountName":"admin",
      "accountUID":"3f74a85e39e64432ba917a2e60fa15aa",
      "accountPassword":"GJN8p2ol"
   }
}
```

Where:

- **accountUID** is the account's unique identifier.
- **accountName** is the name of the account.
- **accountPassword** is the account password.

B.7.22 Show Password History

Use this API to retrieve and display the password history associated with an account.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role or you must have checked out the account to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}/showpasswordhistory
- **Method:** `GET`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of account token

Example B–62 Sample JSON Representation of Account Token

```
{
  "accountName": "opamuser1",
  "accountUID": "c1b054ed0f984e27bd68b8c28b985801",
  "passwordHistory": [
    {
      "accountPassword": "M7aGfNOR",
      "modificationTime": "1382996686"
    },
    {
      "accountPassword": "Dr3z5AGa",
      "modificationTime": "1382996412"
    }
  ]
}
```

Where:

- **accountUID** is the account's unique identifier.
- **accountName** is the name of the account.
- **passwordHistory** is the account password history.
- **accountPassword** is the account password.
- **modificationTime** is the time (in UTC seconds) when the password was modified.

B.7.23 Show Password History (*Deprecated*)

> **Note:** This API has been deprecated. Oracle recommends that you use the Show Password History API in Section B.7.22, "Show Password History."

Use this API to retrieve and display the password history associated with an account.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role or you must have checked out the account to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}/showpasswordhistory
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of account token

Example B–63 Sample JSON Representation of Account Token

```
{
    "accountName":"admin",
    "accountUID":"3f74a85e39e64432ba917a2e60fa15aa",
    "passwordHistory": [{
            "accountPassword": "Ud2fykRx",
            "modificationTime": "2013-01-27 19:36:32.952"
        }, {
            "accountPassword": "jgs21Z8w",
            "modificationTime": "2013-01-27 19:37:02.449"
        }, {
            "accountPassword": "I3jDRaZb",
            "modificationTime": "2013-01-27 19:37:19.488"
        }, {
            "accountPassword": "5VfKaYZT",
            "modificationTime": "2013-01-28 00:22:37.331"
    }]
}
```

Where:

- **accountUID** is the account's unique identifier.
- **accountName** is the name of the account.
- **passwordHistory** is the account password history.
- **accountPassword** is the account password.
- **modificationTime** is the time when the password was modified.

B.7.24 Reset Password

Use this API to reset the password on the account.

Note: You must be an administrator with the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/account/{accountUID}/resetpassword
- **Method:** `GET`
- **Content-Type:** application/json
- **Body:** JSON representation of the new password
- **Returns on Success:** Status code 200

Example B–64 Sample JSON Representation of the New Password

```
{
   "password":"password1"
}
```

Or,

```
{
   "autogen":"true"
}
```

Where:

- **accountUID** is the account's unique identifier.
- **password** is the password assigned to the account.
- **autogen** is the a flag that controls whether to generate a password automatically or not. The default value if this flag is `false`.

B.8 UI Resource

The APIs described in this section include:

- Section B.8.1, "Search Accounts (Deprecated)"
- Section B.8.2, "Search Assigned Accounts (Deprecated)"
- Section B.8.3, "Get All Checked Out Accounts (Deprecated)"
- Section B.8.4, "Retrieve Checked-Out Accounts or Checkout Distribution"
- Section B.8.5, "Retrieve Checked-Out Account Information"

B.8.1 Search Accounts (*Deprecated*)

Note: This API has been deprecated. Oracle recommends that you use the Search Accounts API in Section B.7, "Account Resource."

Use this API to search accounts using one or more of the following search request parameters:

- `type`
- `domain`

- description
- name
- accountname

All of these parameters are *optional*.

> **Note:** You must be an administrator with the *User Manager* Admin Role or the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/ui/allaccounts/search?param1=val1&param2=val2
- **Method:** GET
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of account collection

Example B–65 Sample JSON Representation of Account Collection

```
{
    "AccountCollection" : [
        {
            "account" : {
            "shared" : false,
            "targetUID" : "62bcfb98f95174ad1900ea2535989b53",
            "domain" : "needtofix",
            "targetName" : "sunds_6.3_target",
            "targetType" : "ldap",
            "accountlevelstatus" : "checkedIn",
            "description" : "",
            "accountName" : "dsperson1",
            "uri" : "https://localhost:7002/opam/account/35e2709edf0443edae8f67727d937bec",
            "accountUID" : "35e2709edf0443edae8f67727d937bec"
        }
            },
        {
            "account" : {
            "shared" : false,
            "targetUID" : "62bcfb98f95174ad1900ea2535989b53",
            "domain" : "needtofix",
            "targetName" : "sunds_6.3_target",
            "targetType" : "ldap",
            "accountlevelstatus" : "checkedIn",
            "description" : "",
            "accountName" : "dsperson10",
            "uri" : "https://localhost:7002/opam/account/0a1ee2cb17e345cdb537a2f05e11e93c",
            "accountUID" : "0a1ee2cb17e345cdb537a2f05e11e93c"
            }
        }

    ],
    "count" : 2
}
```

For all other attribute definitions, refer to Section B.6, "Target Resource" and Section B.7, "Account Resource."

B.8.2 Search Assigned Accounts (*Deprecated*)

Note: This API has been deprecated. Oracle recommends that you use the Section B.7.6, "Search Assigned Accounts" API in Section B.7, "Account Resource."

Use this API to search assigned accounts using one or more of the following search request parameters:

- `type`
- `domain`
- `description`
- `name`
- `accountname`

All of these parameters are *optional*.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/ui/myaccounts/search?param1=val1&param2=val2
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of account collection

Example B–66 Sample JSON Representation of Account Collection

```
{
    "AccountCollection" : [
        {
            "account" : {
                "status" : "checkedIn",
                "shared" : false,
                "targetUID" : "62bcfb98f95174ad1900ea2535989b53",
                "domain" : "needtofix",
                "targetName" : "ldap1_target",
                "targetType" : "ldap",
                "accountlevelstatus" : "checkedIn",
                "description" : "",
                "accountName" : "person1",
                "uri" : "https://localhost:7002/opam/account/0d755f646bcf4fa08ca515ed3829aadf",
                "accountUID" : "0d755f646bcf4fa08ca515ed3829aadf"
        }
            },
        {
                "account" : {
                "status" : "checkedIn",
                "shared" : false,
                "targetUID" : "62bcfb98f95174ad1900ea2535989b53",
```

```
                "domain" : "needtofix",
                "targetName" : "ldap1_target",
                "targetType" : "ldap",
                "accountlevelstatus" : "checkedIn",
                "description" : "",
                "accountName" : "person2",
                "uri" : "https://localhost:7002/opam/account/62c684c3821f4e118790e815ee881e02",
                "accountUID" : "62c684c3821f4e118790e815ee881e02"
            }
        }
    ],
    "count" : 2
}
```

Where "**status**" indicates whether the requesting user has checked out the account or not.

For all other attribute definitions, refer to Section B.6, "Target Resource" and Section B.7, "Account Resource."

B.8.3 Get All Checked Out Accounts (*Deprecated*)

Note: This API has been deprecated. Oracle recommends that you use the Get All Checked Out Accounts API in Section B.7, "Account Resource."

Use this API to retrieve a list of all accounts that have been checked out by the logged in user.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/ui/allaccounts/mycheckedout
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of account collection

Example B–67 Sample JSON Representation of Account Collection

```
{
   "AccountCollection":[
      {
         "account":{
            "uri":"https:\/\/opam_server_host:opam_ssl_port\/opam\/account\
               /3740553e999a4f6aa8e8f9286d320cb4",
            "accountUID":"3740553e999a4f6aa8e8f9286d320cb4",
            "accountName":"sherlock",
            "status":"checkedOut",
            "targetUID":"62bcfb98f95174ad1900ea2535989b53",
            "targetName":"ldap1-target",
            "targetType":"ldap",
            "domain":"berkeley",
            "expiryTime":1338765551,
         },
         "count":1
      }
```

```
   ]
}
```

For attribute definitions, refer to Section B.6, "Target Resource" and Section B.7, "Account Resource."

B.8.4 Retrieve Checked-Out Accounts or Checkout Distribution

Use this API to retrieve the checked-out accounts distribution or retrieve checkout distribution.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/ui/report/usage/checkedoutdistribution? para1=value1&para2=value2&para3=value3
- **Method:** `GET`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of checked out accounts distribution

Example B–68 Example JSON Output of Checked Out Accounts Distribution

```
{
  "CheckoutDistribution": [
    {
      "ldap": 3
    },
    {
      "lockbox": 2
    },
    {
      "unix": 1
    }
  ]
}
```

Where:

- **org** is the target organization. For example, enter `org=ldap` to search only LDAP organizations, or enter `org = null` to search all organizations.
- **type** is the checkout type. For example, enter
  - **type = password checkout** for password checkout distribution.
  - **type = session checkout** for session checkout distribution.
  - **type =null** defaults to password checkout.
- **dis_type** is the distribution type. For example, enter
  - **dis_type = checkout** for password or session checkout distribution.
  - **dis_type = account** (Default) for account distribution. If you specify this parameter, then Oracle Privileged Account Manager automatically ignores the (preceding) **type** parameter.

B.8.5 Retrieve Checked-Out Account Information

Use this API to retrieve checked-out account information.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/ui/report/usage/checkedoutaccounts
- **Method:** `GET`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of checked out checkout distribution

Example B–69 Example JSON Output of Checked Out Account Information

```
{
   "CheckedoutAccountCollection":
   [
       {
           "accountUID": "62bcfb98f95174ad1900ea2535989b53",
           "targetUID": "62bcfb98f95174ad1900ea2535989b53",
           "accountName": "OPAM_DB_ACC1",
           "targetName": "OPAM_DB_ACC1",
           "targetType": "database",
           "host": "adc6170584"
       },
       {
           "accountUID": "a044bd2aec7de5d70d73f97645db9191",
           "targetUID": "a044bd2aec7de5d70d73f97645db9191",
           "accountName": "cluser1",
           "targetName": "cluser1",
           "targetType": "ldap",
           "host": "myhost.us.example.com"
       }
   ]
}
```

B.9 User Resource

The APIs described in this section include:

- Section B.9.1, "Get a User"
- Section B.9.2, "Get All Accounts Granted to a User"
- Section B.9.3, "Search Users from Identity Store"
- Section B.9.4, "Search for Assigned Users"

B.9.1 Get a User

Use this API to retrieve a user.

> **Note:** You must be an administrator with the *User Manager* Admin Role or the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/user/{uid}
- **Method:** `GET`

- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of user

Example B–70 Sample JSON Representation of User

```
{
   "user":{
      "uid":"opamuser1",
      "lastname":"opamuser1",
      "usertype":"End-User",
      "opamrole":[

      ],
      "dn":"uid=opamuser1,ou=people,ou=myrealm,dc=base_domain",
   }
}
```

Where:

- **uid** is the login ID of the user.
- **lastname** is the last name of the user.
- **firstname** is the first name of the user.
- **dn** is the distinguished name of the user.
- **usertype** indicates whether the user has an Administrative Role.
- **opamrole** is the user's Admin Role.

B.9.2 Get All Accounts Granted to a User

Use this API to retrieve all of the accounts granted to a user.

> **Note:** You must be an administrator with the *User Manager* Admin Role or the *Security Administrator* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/user/{uid}/accounts
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of accounts collection

Example B–71 Sample JSON Representation of Accounts Collection

```
{
    "accounts":[
        {
            "account":{
            "accountUID":"16d245784350469cbe25229a7c45af22",
            "accountName":"oidperson10",
            "targetID":"75a23e9f30ba456b961a1f5d327e67ef",
            "targetName":"ldap1_target",
            "targetDomain":"needtofix",
```

```
            "targetType":"ldap"
        }
    },
        {
            "account":{
            "accountUID":"47671a7a4ebc44c496888aac5423dad1",
            "accountName":"oudperson11",
            "targetID":"488d6d656b2c4b96a5fd835c131b4c00",
            "targetName":"oud_11.115_target",
            "targetDomain":"needtofix",
            "targetType":"ldap"
        }
    }
    ]
}
```

For attribute definitions, refer to Section B.6, "Target Resource" and Section B.7, "Account Resource."

B.9.3 Search Users from Identity Store

Use this API to search for users. This API searches for the `searchKeyWord` in firstname, lastname, uid, and mail of the user.

> **Note:** You must be an administrator with the *User Manager* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/user/search/{searchKeyWord}
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of users

Example B–72 Sample JSON Representation of Users

```
{
   "users":[
      {
         "user":{
            "uid":"opamenduser1",
            "firstname":"opamenduser1",
            "lastname":"opamenduser1",
            "dn":"uid=opamenduser1,ou=people,ou=myrealm,dc=base_domain"
         }
      },
      {
         "user":{
            "uid":"opamenduser2",
            "lastname":"opamenduser2",
            "dn":"uid=opamenduser2,ou=people,ou=myrealm,dc=base_domain"
         }
      },
      {
```

```
         "user":{
            "uid":"opamuser1",
            "lastname":"opamuser1",
            "dn":"uid=opamuser1,ou=people,ou=myrealm,dc=base_domain"
         }
      }
   ]
}
```

For attribute definitions, refer to Section B.9.1, "Get a User."

B.9.4 Search for Assigned Users

Use this API to search for users. This API is contains a search with the uid parameter.

The uid parameter is *optional*.

> **Note:** You must be an administrator with the *User Manager* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/user/advancedsearch?param1=val1&param2=val2
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of users

Example B–73 Sample JSON Representation of Users

```
{
   "users":[
      {
         "user":{
            "uid":"OracleSystemUser",
            "lastname":"OracleSystemUser",
            "dn":"uid=OracleSystemUser,ou=people,ou=myrealm,dc=base_domain"
         }
      },
      {
         "user":{
            "uid":"weblogic",
         }
      },
      {
         "user":{
            "uid":"app_config",
            "lastname":"app_config",
            "dn":"uid=app_config,ou=people,ou=myrealm,dc=base_domain"
         }
      },
      {
         "user":{
            "uid":"sec_admin",
            "lastname":"sec_admin",
            "dn":"uid=sec_admin,ou=people,ou=myrealm,dc=base_domain"
```

```
            }
         },
         {
            "user":{
               "uid":"user_manager",
               "lastname":"user_manager",
               "dn":"uid=user_manager,ou=people,ou=myrealm,dc=base_domain"
            }
         },
         {
            "user":{
               "uid":"sec_auditor",
               "lastname":"sec_auditor",
               "dn":"uid=sec_auditor,ou=people,ou=myrealm,dc=base_domain"
            }
         },
         {
            "user":{
               "uid":"opamenduser1",
               "firstname":"opamenduser1",
               "lastname":"opamenduser1",
               "dn":"uid=opamenduser1,ou=people,ou=myrealm,dc=base_domain"
            }
         },
         {
            "user":{
               "uid":"opamenduser2",
               "lastname":"opamenduser2",
               "dn":"uid=opamenduser2,ou=people,ou=myrealm,dc=base_domain"
            }
         },
         {
            "user":{
               "uid":"opamuser1",
               "lastname":"opamuser1",
               "dn":"uid=opamuser1,ou=people,ou=myrealm,dc=base_domain"
            }
         }
      ]
}
```


For attribute definitions, refer to Section B.9.1, "Get a User."

B.10 Group Resource

The APIs described in this section include:

- Section B.10.1, "Get Group"
- Section B.10.2, "Get Member Users of a Group"
- Section B.10.3, "Get Member Groups of a Group"
- Section B.10.4, "Get All Accounts Granted to a Group"
- Section B.10.5, "Search Groups from Identity Store"
- Section B.10.6, "Advanced Search for Assigned Groups"

B.10.1 Get Group

Use this API to retrieve a group.

> **Note:** You must be an administrator with the *User Manager* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/group/{name}
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of group

Example B–74 Sample JSON Representation of Group

```
{
   "group":{
      "name":"opamgroup1",
      "dn":"cn=opamgroup1,ou=groups,ou=myrealm,dc=base_domain",
      "description":"",
   }
}
```

Where:

- **name** is the name of the group.
- **dn** is the distinguished name of the group.
- **description** is a description of the group.

B.10.2 Get Member Users of a Group

Use this API to retrieve the user members of a group.

> **Note:** You must be an administrator with the *User Manager* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/group/{name}/users
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of user collection

Example B–75 Sample JSON Representation of User Collection

```
{
    "users":[
        {
            "user":{
            "uid":"master_user",
            "lastname":"master_user",
            "dn":"uid=master_user,ou=people,ou=myrealm,dc=base_domain"
```

```
        }
    },
        {
            "user":{
            "uid":"sec_admin",
            "lastname":"sec_admin",
            "dn":"uid=sec_admin,ou=people,ou=myrealm,dc=base_domain"
        }
    }
    ]
}
```

For attribute definitions, refer to Section B.9.1, "Get a User."

B.10.3 Get Member Groups of a Group

Use this API to retrieve the group members of a group.

> **Note:** You must be an administrator with the *User Manager* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/group/{name}/groups
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of group collection

Example B–76 Sample JSON Representation of Group Collection

```
{
    "groups":[
        {
            "group":{
                "name":"CrossDomainConnectors",
                "description":"CrossDomainConnectors can make inter-domain calls from foreign
domains."
            }
        },
       {
            "group":{
               "name":"Deployers",
               "description":"Deployers can view all resource attributes and deploy applications."
        }
    }
    ]
}
```

For attribute definitions, refer to Section B.10.1, "Get Group."

B.10.4 Get All Accounts Granted to a Group

Use this API to retrieve the all of the accounts granted to a group.

Note: You must be an administrator with the *User Manager* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/group/{name}/accounts
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and representation of accounts collection

Example B–77 Sample JSON Representation of Accounts Collection

```
{
    "accounts":[
        {
            "account":{
            "accountUID":"16d245784350469cbe25229a7c45af22",
            "accountName":"oidperson10",
            "targetID":"75a23e9f30ba456b961a1f5d327e67ef",
            "targetName":"ldap1_target",
            "targetDomain":"needtofix",
            "targetType":"ldap"
            }
        },
        {
            "account":{
            "accountUID":"47671a7a4ebc44c496888aac5423dad1",
            "accountName":"oudperson11",
            "targetID":"488d6d656b2c4b96a5fd835c131b4c00",
            "targetName":"oud_11.115_target",
            "targetDomain":"needtofix",
            "targetType":"ldap"
            }
        }
        ]
}
```

For attribute definitions, refer to Section B.6, "Target Resource" and Section B.7, "Account Resource."

B.10.5 Search Groups from Identity Store

Use this API to search for groups. This API searches for the `searchKeyWord` in the group names.

Note: You must be an administrator with the *User Manager* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/group/search/{searchKeyWord}
- **Method:** `GET`
- **Content-Type:** NA

- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of groups

Example B–78 Sample JSON Representation of Groups

```
{
   "groups":[
      {
         "group":{
            "name":"opamgroup1",
            "description":"",
         }
      },
      {
         "group":{
            "name":"opamgroup2",
            "description":"",
         }
      },
      {
         "group":{
            "name":"opamsubgroup1",
            "description":"",
         }
      },
      {
         "group":{
            "name":"opamsubgroup2",
            "description":"",
         }
      },
      {
         "group":{
            "name":"OPAM_APPLICATION_CONFIGURATOR",
            "description":"OPAM_APPLICATION_CONFIGURATOR",
         }
      },
      {
         "group":{
            "name":"OPAM_SECURITY_ADMIN",
            "description":"OPAM_SECURITY_ADMIN",
         }
      },
      {
         "group":{
            "name":"OPAM_SECURITY_AUDITOR",
            "description":"OPAM_SECURITY_AUDITOR",
         }
      },
      {
         "group":{
            "name":"OPAM_USER_MANAGER",
            "description":"OPAM_USER_MANAGER",
         }
      }
   ]
}
```

For attribute definitions, refer to Section B.10.1, "Get Group."

B.10.6 Advanced Search for Assigned Groups

Use this API to search for groups who have been assigned an account. The request parameter is `groupname`, which is *optional*.

> **Note:** You must be an administrator with the *User Manager* Admin Role to use this API.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/group/advancedsearch?param1=val1&param2=val2.
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of groups

Example B–79 Sample JSON Representation of Groups

```
{
   "groups":[
      {
         "group":{
            "name":"opamgroup1",
            "description":"",
         }
      },
      {
         "group":{
            "name":"opamgroup2",
            "description":"",
         }
      },
      {
         "group":{
            "name":"opamsubgroup1",
            "description":"",
         }
      },
      {
         "group":{
            "name":"opamsubgroup2",
            "description":"",
         }
      },
      {
         "group":{
            "name":"OPAM_APPLICATION_CONFIGURATOR",
            "description":"OPAM_APPLICATION_CONFIGURATOR",
         }
      },
      {
         "group":{
            "name":"OPAM_SECURITY_ADMIN",
            "description":"OPAM_SECURITY_ADMIN",
         }
      },
```

```
      {
         "group":{
            "name":"OPAM_SECURITY_AUDITOR",
            "description":"OPAM_SECURITY_AUDITOR",
         }
      },
      {
         "group":{
            "name":"OPAM_USER_MANAGER",
            "description":"OPAM_USER_MANAGER",
         }
      }
   ]
}
```

For attribute definitions, refer to Section B.10.1, "Get Group."

B.11 Resource Groups Resource

The APIs described in this section include:

- Section B.11.1, "Create a Resource Group"
- Section B.11.2, "Search Resource Groups"
- Section B.11.3, "View a Resource Group"
- Section B.11.4, "Update a Resource Group"
- Section B.11.5, "Delete a Resource Group"
- Section B.11.6, "Create or Delete a Delegation"
- Section B.11.7, "View Delegations on a Resource Group"

B.11.1 Create a Resource Group

Use this API to create a resource group.

Note:

- You must be an administrator with the *Security Administrator* Admin Role to create resource groups.
- If you have been delegated the *Security Administrator* Admin Role, you can use this API to create resource groups under *delegated* resource groups by using the `memberof` option.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/resourcegroup
- **Method:** `POST`
- **Content-Type:** application/json
- **Body:** JSON representation of the resource group to be added
- **Returns on Success:** Status code 200 and JSON Representation of Connector Server
- **Returns on Failure:** NA

Example B–80 Sample JSON Representation of a Resource Group

```
{
    "resourcegroup": {
        "resourcegroupname":"rg1",
        "description":"this is the description of the resource group"
    }
}

{
   "resourcegroup":{
      "resourcegroupname":"rg1",
      "description":"this is a resource group",
      "members":[
         {
            "member":{
               "memberid":"26894a8fadc8448b9bf01eb6f52402c1",
               "membertype":"account"
            }
         },
         {
            "member":{
               "memberid":"fcc6ec6530174fcb94b066e3dc1469e4",
               "membertype":"account"
            }
         },
         {
            "member":{
               "memberid":"0542ef9600e6479a8597ec1f8479276e",
               "membertype":"resourcegroup"
            }
         },
         {
            "member":{
               "memberid":"fc142a16c1c04148923049c8798abfdb",
               "membertype":"target"
            }
         }
      ]
   }
}
```

Where:

- **org** is the target organization (`org = null` is search all organizations.)
- **type** is the password checkout or session checkout. Use this parameter to specify password checkout distribution or session checkout distribution. (`type=null` is default for password checkout.)
- **dis_type** is distribution type, either checked-out account distribution or checkout distribution. (Default is account distribution). You can use dis_type = checkout to choose password or session checkout distribution and dis_type = account automatically ignores the preceding parameter type.

B.11.2 Search Resource Groups

Use this API to search for a resource group.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to search for resource groups.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/resourcegroup/{resourcegroupUID}
- **Method:** NA
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** NA
- **Returns on Failure:** NA

B.11.3 View a Resource Group

Use this API to view a resource group.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role or the *User Manager* Admin Role to view resource groups.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/resourcegroup/{resourcegroupUID}
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of the resource group
- **Returns on Success:** NA

Example B–81 Sample JSON Representation of a Resource Group

```
{
   "resourcegroup":{
      "resourcegroupid":"ae43b6e6dd664ee1b8a547f44f3278e8",
      "resourcegroupname":"rg1",
      "description":"updated description for rg1",
      "members":[
         {
            "member":{
               "memberid":"26894a8fadc8448b9bf01eb6f52402c1",
               "membername":"a1",
               "membertype":"account"
            }
         },
         {
            "member":{
               "memberid":"fcc6ec6530174fcb94b066e3dc1469e4",
               "membername":"a2",
               "membertype":"account"
```

```
            }
         },
         {
            "member":{
               "memberid":"0542ef9600e6479a8597ec1f8479276e",
               "membername":"rg2",
               "membertype":"resourcegroup"
            }
         },
         {
            "member":{
               "memberid":"f433674a53c448f9ae820f12995ba51d",
               "membername":"t1",
               "membertype":"target"
            }
         }
      ]
   }
}
```

B.11.4 Update a Resource Group

Use this API to update a resource group (for example modify the description or add and remove members in the resource group).

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to update resource groups.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/resourcegroup/{resourcegroupUID}
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation of resource group modification
- **Returns on Success:** Status code 200

Example B–82 Sample JSON Representation of a Resource Group Modification

```
{
   "modifications":[
      {
         "modification":{
            "description":"updated description"
         }
      },
      {
         "modification":{
            "members":{
               "operation":"add",
               "memberid":"1001",
               "membertype":"account"
            }
         }
      },
```

```
      {
         "modification":{
            "members":{
               "operation":"delete",
               "memberid":"1001"
            }
         }
      }
   ]
}
```

B.11.5 Delete a Resource Group

Use this API to delete a resource group.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to delete resource groups.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/resourcegroup/{resourcegroupUID}
- **Method:** `DELETE`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200
- **Returns on Failure:** NA

B.11.6 Create or Delete a Delegation

Use this API to create a delegation on a resource group.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to create or delete a resource group delegation.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/resourcegroup/{resourcegroupUID}
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200
- **Returns on Failure** NA

Example B–83 Sample JSON Representation of a Resource Group

```
{
   "modifications":[
      {
```

```
         "modification":{
            "delegations":{
               "operation":"add",
               "delegatee":"opam_user1",
               "delegatee_type":"user",
               "privilege":"security_admin"
            }
         }
      },
      {
         "modification":{
            "delegations":{
               "operation":"add",
               "delegatee":"opam_admins",
               "delegatee_type":"role",
               "privilege":"user_manager"
            }
         }
      },
      {
         "modification":{
            "delegations":{
               "operation":"delete",
               "delegatee":"opam_user2",
               "delegatee_type":"user",
               "privilege":"security_admin"
            }
         }
      }
   ]
}
```

B.11.7 View Delegations on a Resource Group

Use this API to view the delegations on a resource group.

> **Note:** You must be an administrator with the *Security Administrator* Admin Role to view resource group delegations.

- **URI**:
  https://*opam_server_host*:*opam_ssl_port*/opam/resourcegroup/{resourcegroupUID}/
  delegations
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of the resource group
- **Returns on Failure:** NA

Example B–84 Sample JSON Representation of Delegations on a Resource Group

```
{
   "resourcegroup":{
```

```
      "resourcegroupid":"139b052b2d1649aa89964b4bafd2476a",
      "resourcegroupname":"rg2",
      "description":"updated description for rg1",
      "delegations":[
         {
            "delegation":{
               "delegatee":"opam_user1",
               "delegatee_type":"user",
               "privilege":"security_admin"
            }
         },
         {
            "delegation":{
               "delegatee":"opam_user2",
               "delegatee_type":"user",
               "privilege":"security_admin"
            }
         },
         {
            "delegation":{
               "delegatee":"opam_admins",
               "delegatee_type":"role",
               "privilege":"user_manager"
            }
         }
      ]
   }
}
```

B.12 Plug-In Resource

The APIs described in this section include:

- Section B.12.1, "Add Plug-In Configuration"
- Section B.12.2, "Verify Plug-In Configuration"
- Section B.12.3, "Search For Plug-In Configuration"
- Section B.12.4, "Retrieve Plug-In Configuration"
- Section B.12.5, "Update Plug-In Configuration"
- Section B.12.6, "Remove Plug-In Configuration"

B.12.1 Add Plug-In Configuration

Use this API to add a plug-in configuration.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/plugin
- **Method:** `POST`
- **Content-Type:** application/json
- **Body:** JSON representation of plug-in
- **Returns on Success:** Status code 201 and Location
- **Returns on Error:** NA

Example B–85 Sample JSON Representation of Plug-In Configuration Creation

```
{
    "plugin": {
        "pluginName":"sampleplugin"
        "pluginDescription":"Sample Plugin"
        "pluginEnabled": "true"
        "pluginResource":"account"
        "pluginOperation":"checkout"
        "pluginTiming":"post"
        "pluginOrder":"10"
        "pluginClassName":"EmailNotifyPlugin"
        "pluginClassPath":"/u01/plugins/emailplugin.jar"
        "pluginEnableGroup":["hrgroup", "itgroup"]
        "pluginEnableUser":["admin"]
        "pluginEnableResult":"200"
        "pluginVersion":"1.0.0"
        "pluginCustomAttrs":[
            {
                "pluginCustomAttr": {
                    "attrname":"notificationemail"
                    "attrvalue":"abc@abc.com"
                }
            }
        ]
    }
}
```

Sample Output

https://*opam_server_host*:*opam_ssl_port*/opam/plugin/9bbcbbb087174ad1900ea691a2573b61

B.12.2 Verify Plug-In Configuration

Use this API to validate a plug-in configuration, which includes

- Testing the uniqueness of the `pluginName`
- Testing the uniqueness of the `pluginResource`, `pluginOperation`, `pluginOrder` combination
- Validating attributes and allowed values
- Validating the loading of `pluginClassName` using the `pluginClassPath`

The API is as follows:

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/plugin/test
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation of plug-in
- **Returns on Success:** Status code 200
- **Returns on Error:** NA

Example B–86 Sample JSON Representation of Plug-In Configuration for Verification

```
{
    "plugin": {
```

```
        "pluginUID":"9bbcbbb087174ad1900ea691a2573b61"
        "pluginName":"sampleplugin"
        "pluginDescription":"Sample Plugin"
        "pluginEnabled": "true"
        "pluginResource":"account"
        "pluginOperation":"checkout"
        "pluginTiming":"post"
        "pluginOrder":"10"
        "pluginClassName":"EmailNotifyPlugin"
        "pluginClassPath":"/u01/plugins/emailplugin.jar"
        "pluginEnableGroup":["hrgroup", "itgroup"]
        "pluginEnableUser":["admin"]
        "pluginEnableResult":"200"
        "pluginVersion":"1.0.0"
        "pluginCustomAttrs":[
            {
                "pluginCustomAttr": {
                    "attrname":"notificationemail"
                    "attrvalue":"abc@abc.com"
                }
            }
        ]
    }
}
```

B.12.3 Search For Plug-In Configuration

Use this API, with any of the following parameters, to search for plug-in configurations:

- Name
- Description
- Enabled
- Resource
- Operation
- Timing

The API is as follows:

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/plugin/search?param1=value1&param2=value2
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of plug-in collection

Sample URI

https://*opam_server_host*:*opam_ssl_port*/opam/plugin/search?name=email&enabled=true&timing=post

Example B–87 Sample JSON Representation of Plug-In Collection

```
{"pluginCollection": [
    {"plugin": {
```

```
        "pluginUID":"9bbcbbb087174ad1900ea691a2573b61"
        "pluginDescription":"Sample Plugin"
        "pluginName":"sampleplugin"
        "pluginEnabled": "true"
        "pluginResource":"account"
        "pluginOperation":"checkout"
        "pluginTiming":"post"
        "pluginOrder":"10"
        "pluginClassName":"EmailNotifyPlugin"
        "pluginClassPath":"/u01/plugins/emailplugin.jar"
        "pluginEnableGroup":["hrgroup", "itgroup"]
        "pluginEnableUser":["admin"]
        "pluginEnableResult":"200"
        "pluginVersion":"1.0.0"
        "pluginCustomAttrs":[
            {
                "pluginCustomAttr": {
                    "attrname":"notificationemail"
                    "attrvalue":"abc@abc.com"
                }
            }
        ]
    }}
   ]
}
```

B.12.4 Retrieve Plug-In Configuration

Use this API to retrieve a plug-in configuration.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/plugin/plugin/{pluginUID}
- **Method:** `GET`
- **Content-Type:** NA
- **Body:** NA
- **Returns on Success:** Status code 200 and JSON representation of a plug-in

Example B–88 Sample JSON Representation of Plug-In

```
{
    "plugin": {
        "pluginUID":"9bbcbbb087174ad1900ea691a2573b61"
        "pluginName":"sampleplugin"
        "pluginDescription":"Sample Plugin"
        "pluginEnabled": "true"
        "pluginResource":"account"
        "pluginOperation":"checkout"
        "pluginTiming":"post"
        "pluginOrder":"10"
        "pluginClassName":"EmailNotifyPlugin"
        "pluginClassPath":"/u01/plugins/emailplugin.jar"
        "pluginEnableGroup":["hrgroup", "itgroup"]
        "pluginEnableUser":["admin"]
        "pluginEnableResult":"200"
        "pluginVersion":"1.0.0"
        "pluginCustomAttrs":[
            {
```

```
                "pluginCustomAttr": {
                    "attrname":"notificationemail"
                    "attrvalue":"abc@abc.com"
                }
            }
        ]
    }
}
```

B.12.5 Update Plug-In Configuration

Use this API to update a plug-in configuration.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/plugin/{pluginUID}
- **Method:** `PUT`
- **Content-Type:** application/json
- **Body:** JSON representation of a plug-in modification
- **Returns on Success:** Status code 200

Example B–89 Sample JSON Representation to Modify Plug-In

```
{
    "modifications": [
        {
             "modification": {
                 "pluginEnabled":"false"
                 "pluginVersion": "1.0.1"
            }
        }
    ]
}
```

B.12.6 Remove Plug-In Configuration

Use this API to delete a plug-in configuration.

- **URI**: https://*opam_server_host*:*opam_ssl_port*/opam/plugin/{pluginUID}
- **Method:** `DELETE`
- **Content-Type:** application/json
- **Body:** NA
- **Returns on Success:** Status code 200

C

Working with the SSH Connector

This appendix describes Oracle Privileged Account Manager's SSH Connector, and how to work with different SSH targets using the SSH Connector in the following sections:

- Section C.1, "About the SSH Connector"
- Section C.2, "Creating Scripts"
- Section C.3, "Framing the Search Regex"
- Section C.4, "Sample Scripts"

C.1 About the SSH Connector

Network devices such as routers and firewalls can also have privileged accounts that manage the device. Network administrators will require a password management solution to periodically rotate passwords of the accounts and also use password checkout capability to gain access when required.

Network devices that provide SSH interface for communication can be managed with the SSH connector through the Identity Connector Framework (ICF). Using the SSH connector, Oracle Privileged Account Manager can manage the network devices.

Even though network devices provide SSH connectivity, the shell environment and commands used by various targets can be different from each other. Therefore, customizations are required to ensure that the SSH connector can work with different devices.

This appendix describes the use cases in which the SSH Connector is used and how it works with different devices (routers, firewalls, hypervisors, etc) through the SSH connection.

The SSH Connector makes use of scripts to perform operations on the Target. Each script is a simple sequence of commands. Oracle Privileged Account Manager uses these scripts to search for accounts and update passwords of accounts on the network device. Refer to Section C.2, "Creating Scripts" for more information about how to create these scripts.

The network devices, in addition to different commands, also have different formats of listing out the accounts that are available for management within the device. A regular expression needs to be defined for the SSH connector to parse and understand the search results provided by the device. Refer to Section C.3, "Framing the Search Regex" for more information about how to create a regular expression.

This appendix also provides details on how to customize the configuration for the SSH connector. After the customization is complete, the SSH target and accounts can be

added to Oracle Privileged Account Manager. Refer to Section 6.2.2.6, "Configuring the SSH Target Type" for detailed information about configuring the SSH target in Oracle Privileged Account Manager.

C.2 Creating Scripts

A script is a sequence of commands that is executed on the target to perform a desired operation. The script structure for the SSH Connector is a sequence of the `COMMAND`, `EXPECT`, and `ERROR` entries. These tags are not case sensitive, but the order in which they are specified must be precise, as described in the following list:

- **Command:** This is the first entry in the SSH Connector script. It is used to specify the command that must be executed on the remote target.
- **Expect:** This is the second entry in the SSH Connector script. It used to identify the successful execution of the command.

  Executing a command generates some output on the screen. The "EXPECT" section is used to specify the output that signifies the successful execution and completion of the command. Once the output mentioned in the "EXPECT" section is seen, the script moves to the next "COMMAND" in the script. For example, this can be a prompt symbol indicating that the command has finished and is awaiting the next command, or this can be a success message printed by the command.
- **Error:** This is the third entry in the SSH Connector script. It is used to specify any error that may be expected to occur due to unsuccessful execution of commands on a target.

  This entry is mandatory for any command that you specify in the script. Though specifying a value of the "ERROR" entry is optional, the tags of the error that you wish to specify must be provided. If you do not expect an error, then this entry can be specified without any expected value as described in the following example:

  Sample value: `ERROR:`

  In the preceding example, the "ERROR" entry has be specified without any value following the colon(:) punctuation mark.

`$__UID__` and `$__PASSWORD__` are the placeholders for the "user" and "password" variables respectively. These placeholders are replaced by the user name and password dynamically. The SSH Connector supports only these two variables.

> **Note:** There are two underscores (__) before and after the UID and PASSWORD placeholders.

The following examples describe the procedure to write different SSH Connector scripts for a Cisco target:

1. To create a script for the Search Users operation, perform the following steps:
   a. Specify a sequence of commands to execute the Search Users operation in the Cisco router, such as the `enable,` and `show run | in username` commands, as described in the following screenshot:

```
lab-4948-2>enable
Password:
lab-4948-2#show run | in username
username engineer secret 5 $1$y.76$cb0zZ2p53uGZ2S6XJKBG20
username arijeet secret 5 $1$.KSx$28KPwzbVc5e1RFcTOp.p11
username test1 secret 5 $1$H1di$au0.v9gIdXEDN29F9pcCi0
username test$2 secret 5 $1$5Mte$4uc6RwwGbHsjKc5Hr7cAV0
username dbusertest$2 secret 5 $1$pP4A$MYaT1FjoEADM.eIjv490S/
username simpletest secret 5 $1$81c4$ykNQxxcm.Ckva5PvZn7eq/
username johndoe secret 5 $1$HkzR$H/7xtX1V36kZcmPwQCyya/
lab-4948-2#~9~
```

b. You must translate this sequence into a script that can be understood by the connector. To do so, specify the entries and their values as described in the following sample script in a text file and save the .txt file:

```
COMMAND:enable
EXCEPT:#
ERROR:

COMMAND:show run | in username
EXCEPT:#
ERROR:
```

2. To create a script for the Change Password operation, perform the following steps:

a. Specify a sequence of commands to execute the Change Password operation in the Cisco router, such as the `enable`, `config terminal`, and `username <user> secret <password>` commands, as described in the following screenshot:

```
login as: engineer
engineer@144.20.10.175's password:

lab-4948-2>enable
Password:
lab-4948-2#show run | in username
username engineer secret 5 $1$y.76$cb0zZ2p53uGZ2S6XJKBG20
username arijeet secret 5 $1$.KSx$28KPwzbVc5e1RFcTOp.p11
username test1 secret 5 $1$H1di$au0.v9gIdXEDN29F9pcCi0
username test$2 secret 5 $1$5Mte$4uc6RwwGbHsjKc5Hr7cAV0
username dbusertest$2 secret 5 $1$pP4A$MYaT1FjoEADM.eIjv490S/
username simpletest secret 5 $1$81c4$ykNQxxcm.Ckva5PvZn7eq/
username johndoe secret 5 $1$HkzR$H/7xtX1V36kZcmPwQCyya/
lab-4948-2#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
lab-4948-2(config)#username test1 secret 0 password
lab-4948-2(config)#exit
lab-4948-2#
```

b. You must translate this sequence into a script that can be understood by the connector. To do so, specify the entries and their values as described in the following sample script in a text file and save the .txt file:

```
COMMAND:enable
EXCEPT:#
ERROR:

COMMAND:configure terminal
EXCEPT:(config)#
ERROR:

COMMAND:username $__UID__ secret $__PASSWORD__
EXCEPT:(config)#
ERROR:Invalid Password length - must contain 1 to 25 characters. Password
```

```
configuration failed|Checking.
```

Note: In the preceding sample script, multiple expected outputs can be provided for an error. These outputs must be separated by the vertical bar or pipe (|) symbol.

3. Create a .properties file that is used to map the operation code (that is used to identify the operation) and the script location. The .properties file contains the location, which are the absolute paths of the scripts, preceded by their operation ID. The absolute path to the scripts must be provided in the .properties file. The SSH Connector supports the three following operations:

Name	Operation ID	Description
search user	SEARCH_ACCOUNT	This script is used to search for an account on the SSH target.
update password	UPDATE_PASSWORD	This script is used to update the password of an account's password. The operation ID for update account.
update account	UPDATE_ACCOUNT	This script is used only for Cisco devices and it is used to update the privileged mode account password.

The following is an example of the entries in the .properties file used for the Cisco target:

/scripts/cisco-nxos/cisco.properties containing

```
SEARCH_ACCOUNT=/scripts/cisco-nxos/CiscoSearchUser.txt
UPDATE_PASSWORD=/scripts/cisco-nxos/CiscoUpdatePassword.txt
UPDATE_ACCOUNT=/scripts/cisco-nxos/CiscoUpdateAccount.txt
```

Note: For more information about the UPDATE_ACCOUNT operation specific to Cisco, refer to Section C.4.1, "Sample Scripts for a Cisco Router With the NX Operating System."

4. Provide the location of the .properties file in the SSH target configuration for the properties file path parameter. Oracle Privileged Account Manager runtime through the SSH connector will use the .properties file to identify and run the scripts corresponding to the operations.

C.3 Framing the Search Regex

The "Search Regex" operation is used to parse the output buffer of a "search account" command in the "search account" script, and fetch the users, roles, or both.

The command used to search for accounts in each device is different depending on the device, therefore resulting in different outputs. The output contains the list of user accounts and roles available on the device with different formatting depending on the system. The SSH connector uses a regular expression to parse the output and understand the user accounts and roles available on the device.

OPAM will parse each line of the output using the search regex. Patterns are separated with the vertical bar or pipe (`|`) symbol. If a line within an output matches one of the patterns, it will be parsed. The string corresponding to `%u` will be parsed as the user name and the string corresponding to `%r` will be parsed as the role name. If you do not want some lines of the content to be parsed, add them within square brackets `[]`.

For example:

```
username %u privilege %r password|username %u privilege %r secret
```

In this example, `%u` is used to fetch USERNAME and `%r` is used to fetch the ROLE or PRIVILEGE_LEVEL. The pipe symbol separates two different formats of output produced by the same device. The connector first checks each line for the first format, if it does not match, then it checks for the next format mentioned after the pipe symbol.

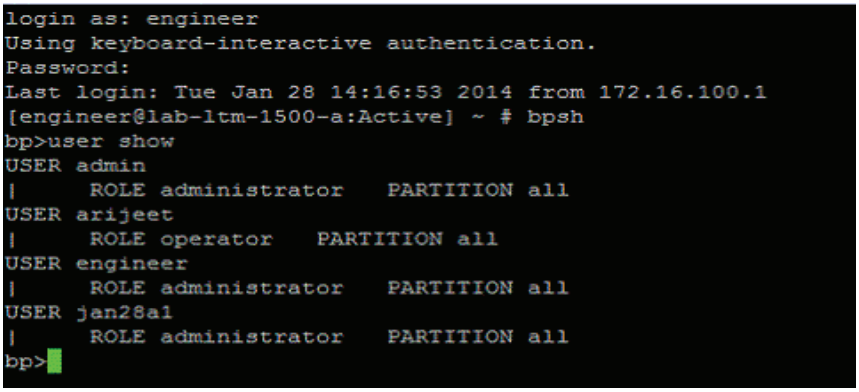
Consider the following cases to frame the Search Regex:

- Case 1: Red Hat Target
- Case 2: Cisco Target
- Case 3: Juniper Target

C.3.1 Case 1: Red Hat Target

In a Red Hat target, when either the user name or the role is present in a single line, the Search Regex is framed as described in this section.

Consider the following example showing the search accounts output from a Red Hat target:



In this example, `USER %u` returns the words after "USER" as the user name, and `ROLE %r PARTITION` returns the words between "ROLE" and "PARTITION" as the role.

Therefore, in a Red Hat target, when either the user name or the role is present in a single line, the Search Regex is framed as "`USER %u|ROLE %r PARTITION`".

C.3.2 Case 2: Cisco Target

In a Cisco target, when both the user name and the role is present in a single line, the Search Regex is framed as follows:

Note: A single target may present the output in different formats. So, to support all the formats for a target, each regex pattern is appended by a vertical bar (|) symbol as shown in the sample pattern in this example.

Consider the following examples showing the search accounts output from a Cisco target:

Example 1:

```
login as: engineer
engineer@144.20.10.175's password:

lab-4948-2>enable
Password:
lab-4948-2#show run | in username
username engineer secret 5 $1$nG7I$ASNidSkvBJiFWIOK9OVST.
username arijeet secret 5 $1$jHyc$UJjhYu7rsXC3hNNDUZhqx0
username jan24C1 privilege 10 secret 5 $1$bPzh$O3xo1rYP/L.s44Bo78CaD0
username jan24C2 privilege 5 secret 5 $1$B7pu$y8e7t7Yja8DGbmFN/NDpU.
lab-4948-2#
```

Example 2:

```
lab-4948-2#show run | in username
username engineer secret 5 $1$nG7I$ASNidSkvBJiFWIOK9OVST.
username arijeet secret 5 $1$jHyc$UJjhYu7rsXC3hNNDUZhqx0
username jan24C1 privilege 10 secret 5 $1$bPzh$O3xo1rYP/L.s44Bo78CaD0
username jan24C2 privilege 5 secret 5 $1$B7pu$y8e7t7Yja8DGbmFN/NDpU.
username checkuser privilege 5 password 0 pass
lab-4948-2#
```

In these examples, `username %u privilege %r password` returns the string between "username" and "privilege" as the user name, and the string between "privilege" and "password" as the role.

Therefore, in a Cisco target, when the user name and the role is present in a single line, the Search Regex is framed as "`username %u privilege %r password|username %u privilege %r secret`".

C.3.3 Case 3: Juniper Target

In a Juniper target, some sections of the output may need to be excluded using the exclude tag `[ ]`. This tag is used to filter the undesirable parts of the output buffer. It will exclude all the lines that include the words specified within the exclude tag.

This is an optional field and must be mentioned at the beginning of the regex if needed. This will exclude the extra labels and prompts from appearing as part of the result. Multiple exclude parameters are separated by a slash (/) symbol. In this example, it is used when there are no words before or after, to fetch the desired values.

Consider the following example showing the search accounts output from a Juniper target:

```
adc-lab-fw-> get admin user
Name                                Privilege
---------------------------------- ---------------
admin                               Root
inet-hq02-root                      Read-Only
jan27u1                             Read-Only
arijeet                             Read-Write
adc-lab-fw->
```

In this example, `USER %u` returns the words after "USER" as the user name, and `ROLE %r PARTITION` returns the words between "ROLE" and "PARTITION" as the role.

The Search Regex in a Juniper target is framed as "`[Name/adc-lab/----]|%u %r`".

C.4 Sample Scripts

The following sections provide sample scripts for various targets:

- Section C.4.1, "Sample Scripts for a Cisco Router With the NX Operating System"
- Section C.4.2, "Sample Scripts for a Juniper Router With the M7I Operating System"
- Section C.4.3, "Sample Scripts for Oracle Integrated Lights Out Manager (ILOM)"

C.4.1 Sample Scripts for a Cisco Router With the NX Operating System

This section provides a table with sample values for JSON and sample scripts for the Cisco target.

> **Note:** Refer to Section 6.2.2.6, "Configuring the SSH Target Type" for detailed information about configuration parameters.

Table C–1 Sample Values For the Configuration Parameters for the Cisco Router

Parameter Name	Sample Values
targetType	SSH
targetName	Cisco-nxos-router
PasswordPolicy	Default Password Policy
passwordrollover	false
managePrivilegeModePassword	false
Host	host
Domain	IT
Description	Cisco Router
Port	22
propertiesFilePath	/scripts/cisco-nxos/ciscoscript.properties

Table C–1 (Cont.) Sample Values For the Configuration Parameters for the Cisco Router

Parameter Name	Sample Values
loginShellPrompt	[$#%>~]
searchResultRegex	username %u password | username %u secret | username %u privilege | username %u
privilegeModePassword	password
loginUserpassword	loginpassword
loginUser	username

C.4.1.1 Contents Of the Script Files

The "SEARCH_ACCOUNT," "UPDATE_PASSWORD," and "UPDATE_ACCOUNT" operations can be configured in the "ciscoscript.properties" file. You must change the path of these three files in the cisco.properties file and provide the absolute path of these files in your environment.

The following sections provide sample content of the .txt files for these operations:

- Contents of the CiscoSearchUser.txt File
- Contents of the CiscoUpdatePassword.txt File
- Contents of the CiscoUpdateAccount.txt File

Contents of the CiscoSearchUser.txt File

```
COMMAND:show run | in username
EXPECT:[>#]
ERROR:
```

Contents of the CiscoUpdatePassword.txt File

```
COMMAND:config terminal
EXPECT:\(config\)#
ERROR:

COMMAND:username $__UID__ password $__PASSWORD__
EXPECT:\(config\)#
ERROR:password is weak

COMMAND:exit
EXPECT:[#]
ERROR:
```

Contents of the CiscoUpdateAccount.txt File

```
COMMAND:enable
EXPECT:#
ERROR:Password:|Bad secrets

COMMAND:config terminal
EXPECT:\(config\)#
ERROR:

COMMAND:enable secret $__ENABLEPASSWORD__
EXPECT:\(config\)#
ERROR:
```

```
COMMAND:exit
EXPECT:#
ERROR:
```

C.4.2 Sample Scripts for a Juniper Router With the M7I Operating System

This section provides a table with sample values for JSON and sample scripts for the SSH target.

Note: Refer to Section 6.2.2.6, "Configuring the SSH Target Type" for detailed information about configuration parameters.

Table C–2 Sample Values for the Configuration Parameters for the Juniper Router

Parameter Name	Sample Values
targetType	SSH
targetName	Juniper-m7i-router
PasswordPolicy	Default Password Policy
passwordrollover	false
managePrivilegeModePassword	false
Host	host
Domain	IT
Description	Juniper Router
Port	22
propertiesFilePath	/scripts/juniper_m7i/juniperscript.properties
loginShellPrompt	[$#%>~]
searchResultRegex	set system login user %u uid
privilegeModePassword	
loginUserpassword	password
loginUser	username

C.4.2.1 Contents Of the Script Files

The "SEARCH_ACCOUNT" and "UPDATE_PASSWORD" operations can be configured in the "juniperscript.properties" file. The following sections provide sample content in the .txt files for these operations:

- Contents of the JuniperSearchUser.txt File
- Contents of the JuniperUpdatePassword.txt File

Contents of the JuniperSearchUser.txt File

```
COMMAND:show configuration | display set | match uid
EXPECT:[>]
ERROR:
```

Contents of the JuniperUpdatePassword.txt File

```
COMMAND:configure
EXPECT:#
ERROR:

COMMAND:set system login user $__UID__ authentication plain-text-password
EXPECT:New password:
ERROR:

COMMAND:$__PASSWORD__
EXPECT:Retype new password:
ERROR:error: require change of case, digits or punctuation

COMMAND:$__PASSWORD__
EXPECT:#
ERROR:

COMMAND:commit
EXPECT:#
ERROR:

COMMAND:exit
EXPECT:>
ERROR:
```

C.4.3 Sample Scripts for Oracle Integrated Lights Out Manager (ILOM)

This section provides a table with sample values for JSON and sample scripts for the Oracle Integrated Lights Out Manager (ILOM).

> **Note:** Refer to Section 6.2.2.6, "Configuring the SSH Target Type" for detailed information about configuration parameters.

Table C–3 Sample Values for the Configuration Parameters for the ILOM Target Type

Parameter Name	Sample Values
targetType	SSH
targetName	ILOM
PasswordPolicy	Default Password Policy
passwordrollover	false
managePrivilegeModePassword	false
Host	host
Domain	IT
Description	ILOM Target
Port	22
propertiesFilePath	/scripts/ILOM/ILOMscript.properties
loginShellPrompt	[$#%>~]

Table C–3 (Cont.) Sample Values for the Configuration Parameters for the ILOM Target Type

Parameter Name	Sample Values
searchResultRegex	[SP/Targets:/->]|%u
privilegeModePassword	
loginUserpassword	password
loginUser	username

C.4.3.1 Contents Of the Script Files

The "SEARCH_ACCOUNT" and "UPDATE_PASSWORD" operations can be configured in the "ILOMscript.properties" file. The following sections provide sample content in the .txt files for these operations:

- Contents of the ILOMSearchUser.txt File
- Contents of the ILOMUpdatePassword.txt File

Contents of the ILOMSearchUser.txt File

```
COMMAND:show -d targets /SP/users
EXPECT:->
ERROR:
```

Contents of the ILOMUpdatePassword.txt File

```
COMMAND:set /SP/users/$__UID__ password=$__PASSWORD__
EXPECT:Enter
ERROR:set: Password length must be between 8 and 16 characters

COMMAND:username $__UID__ password $__PASSWORD__
EXPECT:\(config\)#
ERROR:password is weak

COMMAND:$__PASSWORD__
EXPECT:->
ERROR:
```

Glossary

This glossary contains terms that are specific to administering Oracle Privileged Account Manager.

account

An account on a target.

ADF

Oracle Application Development Framework. An end-to-end development framework, built on top of the Enterprise Java platform, that provides integrated infrastructure solutions for the various layers of an application and an easy way to develop on top of those layers.

application account

Accounts that applications use, at runtime, to connect to other software systems. These accounts are also known as application-to-application authentication.

Application Configurator

Administrative role with privileges to configure and manage Oracle Privileged Account Manager servers.

Authentication provider

A security provider that manages and enforces authentication rules.

For more detailed information, refer to "Configuring Authentication Providers" in the *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

BI Publisher

An Oracle reporting product that can create and manage formatted reports from different data sources.

bootstrap user

A default administrator (`weblogic` user) who is a member of the Administrators group. This user can create and assign users to Oracle Privileged Account Manager Admin Roles and can map users from the domain identity store to Oracle Privileged Account Manager Common Admin Roles.

Credential Store Framework

Refer to ***CSF***.

CRUD

Create, Read, Update, and Delete. Basic functions of persistent storage or a database.

CSF

Credential Store Framework. An OPSS component that primarily provides secure storage for credentials.

DOMAIN_HOME

An environment variable that is usually

MW_HOME/user_projects/domains/<*domain_name*>

Grantee

A user, group, or role that has been granted access to a *privileged account*.

ICF

Identity Connector FrameWork. A component that provides basic provisioning, reconciliation, and other functions required by all Oracle Identity Manager and Oracle Waveset connectors.

Identity Connector FrameWork

Refer to ***ICF***.

identity propagation

Process in which the OPSS Trust Service Asserter examines and validates a token, and then asserts that the identity performing a RESTful call against the Oracle Privileged Account Manager server is the one contained in the token.

JSON representation

JavaScript Object Notation. A lightweight, human-readable data format that is taken from JavaScript and used to exchange information between a browser and a server.

ldifmigrator tool

Oracle Internet Directory Data Migration Tool. Converts LDIF files output from other directories or application-specific repositories into a format recognized by Oracle Internet Directory.

lockbox targets

A target type that does not interact with Oracle Privileged Account Manager, but still provides a secure mechanism for storing the passwords associated with privileged accounts in a deployment.

Oracle Privileged Account Manager client

Component that resides with the Oracle Privileged Account Manager target to provide passwords to the system for unattended connections.

Oracle Privileged Account Manager target

Component that has its privileged passwords managed by Oracle Privileged Account Manager.

OPSS

Oracle Platform Security Services. A standards-based, portable, integrated, enterprise-grade security framework for Java Standard Edition (Java SE) and Java Enterprise Edition (Java EE) applications.

Oracle Application Development Framework

Refer to *ADF*.

Oracle Internet Directory Data Migration Tool

Refer to *ldifmigrator tool*.

Oracle Platform Security Services

Refer to *OPSS*.

Password Policy

Captures the password construction requirements enforced by a specific *target* on an associated *privileged account*. Administrators use this policy to construct the password value that Oracle Privileged Account Manager uses to reset a password on a privileged account. Every privileged account managed by Oracle Privileged Account Manager has an associated Password Policy.

privileged accounts

Accounts on a target that are deemed "privileged" in a deployment and are under Oracle Privileged Account Manager's purview. Accounts are usually privileged when

- They are associated with elevated privileges
- They are used by multiple end-users on a task-by-task basis
- Their use must be controlled and audited

Repository Creation Utility

Oracle Repository Creation Utility. An application that you can use to create a schema and load a repository into the database.

Representational State Transfer

Refer to *REST*.

resources

Representation of targets and accounts.

REST

Representational State Transfer. Software architecture style for distributed hypermedia systems like the World Wide Web. Conforming to REST constraints is otherwise known as being *RESTful*.

run-time password repository

Location from where applications consume the passwords that are required for run-time authentication against other software systems.

SAML

Security Assertion Markup Language. An XML-based open standard product provided by the OASIS Security Services Technical Committee that enables the exchange of authentication and authorization data between security domains.

Security Assertion Markup Language

Refer to *SAML*

service account

An account that Oracle Privileged Account Manager uses when it connects to a target system and to perform all Oracle Privileged Account Manager-related operations (such as discovering accounts, resetting passwords, and so forth) on that target system, Service accounts require some special privileges and properties. Service accounts are sometimes referred to as *unattended accounts*.

shiphome

The directory where you downloaded and extracted Oracle Privileged Account Manager.

target

A software system that contains, uses, and relies on accounts (user, system, or application).

unattended accounts

Refer to ***service account***.

Usage Policy

Defines the constraints around when and how a grantee can use a privileged account. Each privileged account managed by Oracle Privileged Account Manager has an associated Usage Policy.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W