Oracle® Database

Enterprise User Security Administrator's Guide

23ai
F47008-02
May 2024

ORACLE

Oracle Database Enterprise User Security Administrator's Guide, 23ai

F47008-02

Primary Author: Gunjan Jain

Contributors: Rod Ward, Tanvir Ahmed, Chi Ching Chui, Santanu Datta, Janis Greenberg, Rishabh Gupta, Pat Huey, Min-Hank Ho, Yong Hu, Sudha Iyer, Sumit Jeloka, Supriya Kalyanasundaram, Srinidhi Kayoor, Lakshmi Kethana, Manoj Kamani, Van Le, Nina Lewis, Stella Li, Chao Liang, Gopal Mulagund, Sarma Namuduri, Janaki Narasinghanallur, Hozefa Palitanawala, Eric Paapanen, Vikram Pesati, Andy Philips, Richard Smith, Deborah Steiner, Srividya Tata, Philip Thornton, Ramana Turlapati, Sudheesh Varma, Anand Verma, Peter Wahl , Alan Williams

Contents

2 Getting Started with Enterprise User Security

3 Configuration and Administration Tools Overview

4 Enterprise User Security Configuration Tasks and Troubleshooting

5 Administering Enterprise User Security

6 Enterprise User Security Manager (EUSM) Command Reference

List of Examples

List of Figures

List of Tables

Preface

Welcome to the Oracle Database Enterprise User Security Administrator's Guide for Oracle Database.

Oracle Database contains a comprehensive suite of security features that protect your data. These features include database privileges, roles, and integration with the Oracle Identity Management infrastructure for identity management services. Identity management refers to the process by which the complete security lifecycle—account creation, suspension, modification, and deletion—for network entities is managed by an organization.

The Oracle Database Enterprise User Security Administrator's Guide describes how to implement, configure, and administer Oracle Database users in Oracle Internet Directory, a directory service provided by the Oracle Identity Management platform.

This preface contains these topics:

Intended Audience

The Oracle Database Enterprise User Security Administrator's Guide is intended for security administrators, DBAs, and application developers who perform one or more of the following tasks:

- Manage database users and privileges
- Provision database users
- Develop PL/SQL applications for enterprise users

To use this document, you need a working knowledge of SQL and Oracle fundamentals. You should also be familiar with Oracle security features described in "Related Documents".

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, see these Oracle resources:

- *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*
- *Oracle Fusion Middleware Reference for Oracle Identity Management*
- *Oracle Database Advanced Security Guide*
- *Oracle Database 2 Day DBA*
- *Oracle Database Administrator's Guide*
- *Oracle Database Security Guide*
- *Oracle Database Development Guide*
- *Oracle Database SQL Language Reference*
- *Oracle Database Error Messages Reference*
- *Oracle Database Reference*
- *Oracle Database Heterogeneous Connectivity User's Guide*
- *Oracle Database Net Services Administrator's Guide*

Many of the examples in this book use the sample schemas, which are installed by default when you select the Basic Installation option with an Oracle Database installation. Refer to *Oracle Database Sample Schemas* for information on how these schemas were created and how you can use them yourself.

To download free release notes, installation documentation, technical briefs, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

```
Community - Get Started
```

If you already have a user name and password for OTN, then you can go directly to the documentation section of the OTN Web site at

```
Oracle Documentation
```

For conceptual information about the security technologies supported by Enterprise User Security, you can refer to the following third-party publications:

- *Oracle Database 12c Security* by Scott Gaetjen, David Knox, and William Maroulis. McGraw-Hill Education (Publisher), 2015.
- *Oracle Database 12c Security Cookbook* by Zorin Pavlovic and Maja Veselica. Birmingham, UK. Packet Publishing Ltd., 2016.
- *Applied Oracle Security: Developing Secure Database and Middleware Environments* by David Knox, Scott Gaetjen, Hamza Jahangir, Tyler Muth, Patrick Sack, Richard Wark, and Byran Wise. McGraw-Hill Companies Inc., 2010.

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.

Changes in This Release for Oracle Database Enterprise User Security Administrator's Guide

This preface contains the changes in this book for Oracle Database 23ai.

Deprecated Features

This section lists the deprecated features in Oracle Database 23ai.

Deprecation of the mkstore wallet management command line tool

The `mkstore` wallet management command line tool is deprecated with Oracle Database 23ai, and can be removed in a future release.
To manage wallets, Oracle recommends that you use the `orapki` command line tool.

Deprecation of Enterprise User Security (EUS)

Enterprise User Security (EUS) is deprecated with Oracle Database 23ai.
Oracle recommends that you migrate to using Centrally Managed Users (CMU). This feature enables you to directly connect with Microsoft Active Directory without an intervening directory service for enterprise user authentication and authorization to the database. If your Oracle Database is in the cloud, you can also choose to move to one of the newer integrations with a cloud identity provider.

Deprecation of Oracle Virtual Directory with Real Application Security

The use of Oracle Virtual Directory with Oracle Real Application Security is deprecated with Oracle Database 23ai.
Using OVD with Oracle Real Application Security is deprecated, because OVD is no longer updated as a separate product

Desupported Features

This section lists the desupported features in Oracle Database release 23ai release.

Desupport of Diffie-Hellman Anonymous Ciphers

The use of Diffie-Hellman anonymous ciphers (DH anon) is desupported with Oracle Database 23ai for both outbound connections and for database client/server connections. Removing the DH anon ciphers improves the security for Oracle Database connections.

Desupport of Unix Crypt (or MD5crypt) Password Verifier

The Unix Crypt (`MD5crypt`) password verifier algorithm is desupported in Oracle Database 23ai server and clients.

Enterprise User Security (EUS) customers with users in Oracle Internet Directory (OID) potentially can be using older, less secure password verifiers generated by Unix Crypt, either by OID, or by the operating system, before they were migrated to OID. Compared to current methods to hash the password, Unix Crypt is a less secure algorithm. Oracle Database can no longer authenticate EUS or OID users with the older password verifiers. Oracle recommends that you reset passwords in OID now, using newer, more secure hashing algorithms.

Desupport of Oracle Database 10G Password Verifier

Starting with Oracle Database 23ai, the 10G database password verifier is desupported.
The database password verifier for Oracle Database 10g, `10G` is no longer supported or available on Oracle Database 23ai. Refer to the database upgrade guide preinstallation chapters for information about how to identify the Oracle Database 10G database password verifiers, and how to update the database user to use the latest and most secure database password verifier cryptography.

Desupport of Oracle Wallet Manager (OWM)

Starting with Oracle Database 23ai, the Oracle Wallet Manager (OWM) is desupported.
Oracle recommends using the `orapki` command line tool to replace OWM.

Desupport of Enterprise User Security User Migration Utility

Starting with Database 23ai, the User Migration Utility (UMU) part of Enterprise User Security (EUS) is desupported.
There is no workaround.

1

Introducing Enterprise User Security

Enterprise User Security is an important component of the Oracle Database. It enables you to address administrative and security challenges for a large number of enterprise database users.

Enterprise users are those users that are defined in a directory. Their identity remains constant throughout the enterprise. Enterprise User Security relies on Oracle Identity Management infrastructure, which in turn uses an LDAP-compliant directory service to centrally store and manage users.

This chapter explains what Enterprise User Security is and how it works, in the following topics:

- Introduction to Enterprise User Security
- About Using Shared Schemas for Enterprise User Security
- Enterprise User Proxy
- Enterprise User Security Deployment Considerations

1.1 Introduction to Enterprise User Security

This overview of Enterprise User Security explains how it benefits an organization and how enterprise users authenticate and access resources across a distributed database system. It contains the following topics:

- The Challenges of User Management
- Enterprise User Security: The Big Picture
- About Enterprise User Security Directory Entries

1.1.1 The Challenges of User Management

Administrators must keep user information up-to-date and secure for the entire enterprise. This task becomes more difficult as the number of applications and users increases. Typically, each user has multiple accounts on different databases, which means that each user must remember multiple passwords. The result is too many passwords for users to remember and too many accounts for administrators to effectively manage.

With thousands of users accessing database accounts, user administration requires substantial resources. Common information used by multiple applications, such as usernames, telephone numbers, and system roles and privileges, is typically fragmented across the enterprise. Such data increasingly becomes redundant, inconsistent, and difficult to manage.

In addition to user and account management problems, these conditions produce security problems as well. For example, any time a user leaves a company or changes jobs, that user's privileges should be changed the same day in order to guard against their misuse. However, large enterprises often have many user accounts distributed over multiple databases, and an administrator may be unable to make such timely changes.

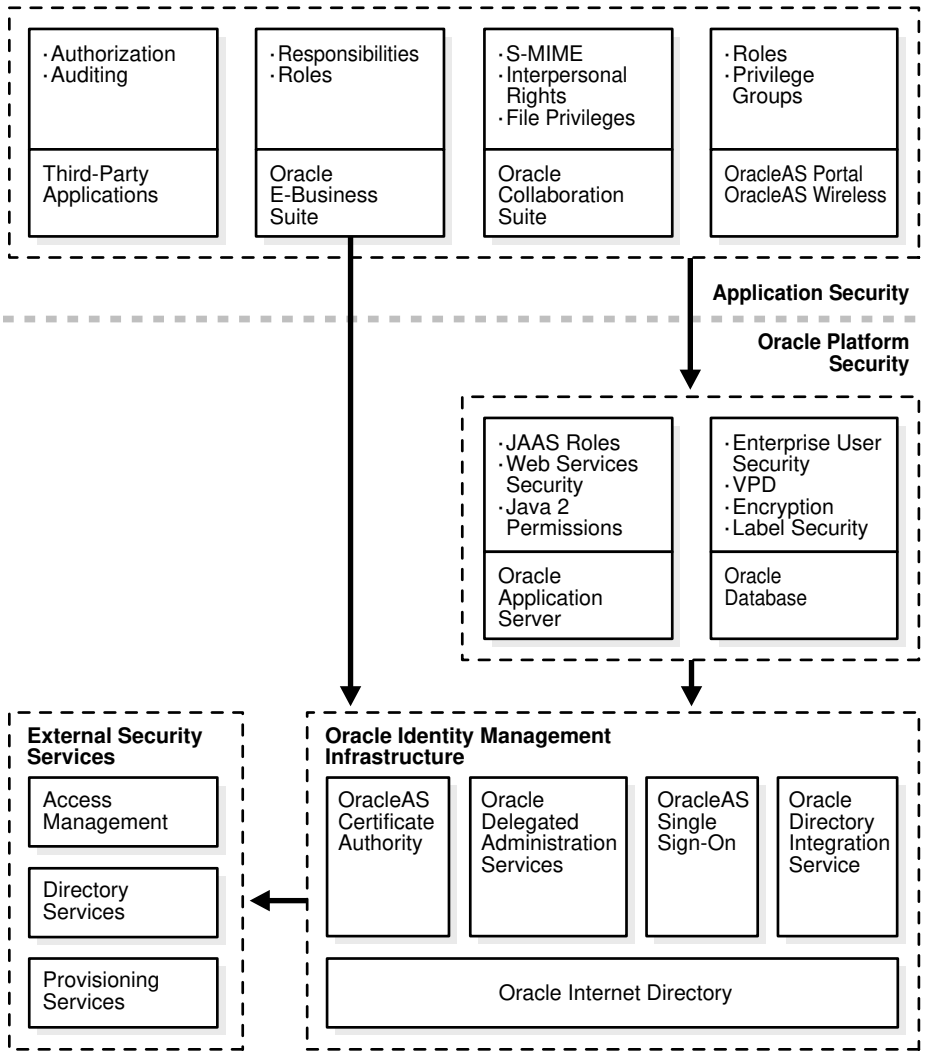
Similarly, if your users have too many passwords, they may write them down, making them easy for others to copy. They may choose passwords that are easy to remember, making them easy for others to guess, and use the same password for multiple applications, risking wider consequences from a compromised password. All such user efforts to track multiple passwords can compromise enterprise security.

1.1.2 Enterprise User Security: The Big Picture

Enterprise User Security addresses user, administrative, and security challenges by relying on the identity management services supplied by Oracle Internet Directory, an LDAP-compliant directory service. Identity management is the process by which the complete security life cycle for network entities is managed in an organization. It typically refers to the management of an organization's application users, where steps in the security life cycle include account creation, suspension, privilege modification, and account deletion.

Figure 1-1 shows how Enterprise User Security fits into the Oracle security architecture, which uses the Oracle Identity Management infrastructure as its foundation.

Figure 1-1 Enterprise User Security and the Oracle Security Architecture



Users benefit from Enterprise User Security through single sign-on (SSO) or single password authentication, depending on the configuration chosen by the administrator. Using single sign-on, users need to authenticate only once and subsequent authentications take place transparently. This functionality requires SSL, which should not be confused with OracleAS Single Sign-On, a component of Oracle Identity Management infrastructure.

Single password authentication lets users authenticate to multiple databases with a single global password although each connection requires a unique authentication. The password is securely stored in the centrally located, LDAP-compliant directory, and protected with security mechanisms including encryption and \. This approach improves usability by reducing the number of passwords to remember and manage, and by eliminating the overhead of setting up SSL.

Enterprise User Security requires Oracle Internet Directory 10g (9.0.4) or higher. Other LDAP-compliant directory services are supported by using Oracle Internet Directory Integration Platform to synchronize them with Oracle Internet Directory. Another directory services product, Oracle Virtual Directory, provides a single, dynamic access point to multiple data sources through LDAP or XML protocols. Oracle Virtual Directory can provide multiple application-specific views of identity data stored in, for example, Oracle Internet Directory, Microsoft Active Directory and Sun Java Systems Directory instances, and can also be used to secure data access to the application-specific sources and enhance high-availability to existing data-sources.

> **See Also:**
>
> *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*, for information about using Oracle Directory Integration Platform and Oracle Virtual Directory with other directories.

> **Note:**
>
> Microsoft Active Directory is supported only for Oracle databases on Windows platforms.

This section contains the following topics:

- How Oracle Internet Directory Implements Identity Management
- Enterprise Users Compared to Database Users
- About Enterprise User Schemas
- How Enterprise Users Access Database Resources with Database Links
- How Enterprise Users Are Authenticated

1.1.2.1 How Oracle Internet Directory Implements Identity Management

Oracle Internet Directory uses the concept of identity management realms to organize information in the directory information tree (DIT), which is a hierarchical tree-like structure consisting of directory object entries. In a directory, each collection of information about an object is called an entry. This object may be a person, but it can also be information about a networked device, such as configuration information. To name and identify the location of

directory objects in the DIT, each entry is assigned a unique distinguished name (DN). The DN of an entry consists of the entry itself and its parent entries, connected in ascending order, from the entry itself up to the root (top) entry in the DIT.

This section contains the following topics:

- About Identity Management Realms
- About Identity Management Realm-Specific Oracle Contexts

1.1.2.1.1 About Identity Management Realms

An identity management realm is a subtree of directory entries, all of which are governed by the same administrative policies. For example, all employees in an enterprise who have access to the intranet may belong to one realm, while all external users who access the public applications of the enterprise may belong to another realm. Use of different realms enables an enterprise to isolate user populations and enforce different administrative policies, such as password policies or naming policies, in each realm. The default nickname attribute, used to identify the login identity, is uid, and it is set in each identity management realm

1.1.2.1.2 About Identity Management Realm-Specific Oracle Contexts

Each identity management realm has a realm-specific Oracle Context (realm Oracle Context) that stores Oracle product information for that realm. A realm Oracle Context stores application data, how users are named and located, how users must be authenticated, group locations, and privilege assignments, all specific to the particular identity management realm in which the realm Oracle Context is located.

> **See Also:**
>
> - *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about Oracle Internet Directory and its architecture
> - About Enterprise User Security Directory Entries for information about Oracle Internet Directory entries that are used for Enterprise User Security

1.1.2.2 Enterprise Users Compared to Database Users

Database users are typically defined in the database by using the `CREATE USER` statement as follows:

```
CREATE USER username IDENTIFIED BY password;
```

This creates a database user, associated with a user schema, who can access the database and be authenticated by using a password with the `CONNECT` command as follows:

```
CONNECT username@database_service_name
Enter Password:
```

Database users must be created in each database they need to access, and they can choose a different password for each database. Database user privileges are controlled by local roles in each database.

In contrast, enterprise users are provisioned and managed centrally in an LDAP-compliant directory, such as Oracle Internet Directory, for database access. Enterprise users have a unique identity in the directory called the distinguished name (DN). When enterprise users log on to a database, the database authenticates those users by using their DN.

Enterprise users are defined in the database as global users. Global users can have their own schemas, or they can share a global schema in the databases they access. You can create enterprise users by using the `GLOBALLY` clause in the `CREATE USER` statement in two different ways.

You can specify a user's directory DN with an `AS` clause, which is shown in the following statement:

```
CREATE USER username IDENTIFIED GLOBALLY AS '<DN of directory user entry>';
```

In this case, they have a schema allocated exclusively to them.

Alternatively, you can specify a null string with the `AS` clause as the following statement shows:

```
CREATE USER username IDENTIFIED GLOBALLY AS '';
```

When you specify a null string with the `AS` clause, the directory maps authenticated users to the appropriate database schema. In this case, multiple users can be mapped to a shared schema based on the mapping information set up and stored in Oracle Internet Directory.

> **Note:**
>
> You can also use the following syntax to create a shared schema:
>
> ```
> CREATE USER username IDENTIFIED GLOBALLY;
> ```
>
> This is the same as specifying a null string.

When enterprise users connect over SSL to the database, they do not use a password. Instead they use the following `CONNECT` command, which looks up the wallet location based on information in the client's `sqlnet.ora` file:

```
CONNECT /@database_service_name
```

Password-authenticated enterprise users use the same `CONNECT` statement to connect to the database as regular database users. For example, password-authenticated enterprise users connect to the database by using the following syntax:

```
CONNECT username@database_service_name
Enter password:
```

When the database receives a connection request from an enterprise user, the database refers to the directory for user authentication and authorization (role) information.

See Also:

- Getting Started with Enterprise User Security for a tutorial on creating and using enterprise users
- "Creating New Enterprise Users"
- *Oracle Database Security Guide* for more information about global users
- *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about defining users in the directory

1.1.2.3 About Enterprise User Schemas

Enterprise users can retain their individual database schemas (exclusive schemas) or share schemas if the enterprise security administrator maps them to a shared schema.

This section contains the following topics:

- Private or Exclusive Schemas
- Shared Schemas

1.1.2.3.1 Private or Exclusive Schemas

If users want to retain their individual schemas in the databases that they access, then they must complete the following tasks:

- Create enterprise users in the directory
- Create a global user schema for each user in each database that they access

Creating separate accounts for each enterprise user on each database that they access results in significant overhead. Instead, creating enterprise users who access a single, generic shared schema in each database increases the efficiency of the enterprise user solution.

1.1.2.3.2 Shared Schemas

To receive the real benefit of the enterprise user solution, you can use shared schemas for your enterprise users. For this strategy, complete the following tasks:

- Create enterprise users in the directory
- Create a single shared schema in each database
- Create a single shared schema mapping in Oracle Internet Directory

Mapping enterprise users to a generic, shared schema on each of the databases that they access greatly reduces the overhead of creating separate schemas for each enterprise user.

Shared schema enterprise users can be mapped to generic, shared schemas on all of the databases that they access, or they can have exclusive schemas on some databases and shared schemas on others. The shared schema mappings are stored in the directory.

> **See Also:**
>
> - "About Using Shared Schemas for Enterprise User Security" for more information about creating and using shared schemas for enterprise users
> - "Creating a Shared Schema in the Database" for a tutorial on creating a shared schema in the database

1.1.2.4 How Enterprise Users Access Database Resources with Database Links

Database links are network objects stored in the local database or in the network definition that identify a remote database, a communication path to that database, and optionally, a user name and password. Once defined, the database link is used to access the remote database. Oracle Database supports connected user links, fixed user links, and current user links.

Enterprise users can use all three types of the following database links:

- Connected user links are accessed by a local user who has an account on the remote server.
- Fixed user links contain a user name and password as part of the link definition.
- Current user database links allow enterprise users to access objects on remote databases without passing authentication information during link execution, or storing authentication information in the link definition.

  They require SSL for the database network connections, which means public key infrastructure (PKI) credentials must be obtained and maintained for the databases. Current user database links can be used to connect to the remote database only as an enterprise user.

> **See Also:**
>
> *Oracle Database Administrator's Guide* for information about all of the different types of database links supported by Oracle Database

1.1.2.5 How Enterprise Users Are Authenticated

Enterprise User Security supports the following authentication methods:

- Password-based authentication
- SSL-based authentication
- Kerberos-based authentication

Each authentication method has advantages and disadvantages. Table 1-1 summarizes the criteria for selecting which authentication method is best for your Enterprise User Security implementation.

Table 1-1 Enterprise User Security Authentication: Selection Criteria

Password Authentication	SSL Authentication	Kerberos Authentication
Password-based authentication	Provides strong authentication over SSL	Provides strong authentication by using Kerberos, version 5 tickets
Provides centralized user and password management	Provides centralized user and PKI credential/wallet management	Provides centralized user and Kerberos credential management
Separate authentications required for each database connection	Supports single sign-on (SSO) using SSL	Supports SSO using Kerberos, version 5 encrypted tickets and authenticators, and authentication forwarding
Retains users' current authentication methods	Initial configuration maybe more difficult because PKI credentials must be generated for all users. (Dependent on administrators' PKI knowledge)	Initial configuration maybe more difficult because Kerberos must be installed and configured to authenticate database users
User identity can be used in two-tier or multitier applications. OracleAS Single Sign-On users and enterprise users use the same stored password	Compatible with either a two-tier or multitier environment	Compatible with either a two-tier or multitier environment
Supports Oracle Release 7.3 and later clients with Oracle Database 10*g* and later	Supports Oracle8*i* and later clients with Oracle Database 10*g* and later	Supports Oracle Database 10*g* and later clients with Oracle Database 10*g* and later
Supports current user database links only if the connection between databases is over SSL	Supports current user database links	Supports current user database links only if the connection between databases is over SSL
Can use third-party directories to store users if synchronized with Oracle Internet Directory[1]	Can use third-party directories to store users if synchronized with Oracle Internet Directory[2]	Can use third-party directories to store users if synchronized with Oracle Internet Directory[3]

1 If third-party directory is Microsoft Active Directory, then when user passwords change, they must be changed in both Active Directory and in Oracle Internet Directory.

2 Must modify the Directory Integration Services agent to synchronize user PKCS #12 attributes.

3 If third-party directory is Microsoft Active Directory, then login to Windows gives you single sign-on login to databases. However, you must modify the Directory Integration Services agent for other third-party directories to synchronize the `KrbPrincipalName` attribute. This synchronization is automatic for Microsoft Active Directory.

> **Note:**
>
> Enterprise User Security supports three-tier environments. Oracle Database proxy authentication features enable
>
> (i) proxy of user names and passwords through multiple tiers, and
>
> (ii) proxy of X.509 certificates and distinguished names through multiple tiers.

> **See Also:**
>
> - Enterprise User Security Configuration Tasks and Troubleshooting for information about configuring the various authentication types for enterprise user security
> - *Oracle Database Security Guide*, for information about using proxy authentication

1.1.3 About Enterprise User Security Directory Entries

In a directory, a collection of information about an object is called an entry. For Enterprise User Security, elements such as users, roles, and databases are directory objects, and information about these objects is stored as entries in the directory.

Each entry in the directory is uniquely identified by a DN. The DN tells you exactly where the entry resides in the directory entry hierarchy, which is commonly called the directory information tree (DIT).

> **Note:**
>
> For Oracle Database 10*g* and later, databases must be registered in a complete identity management realm of Oracle Internet Directory.

> **See Also:**
>
> *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for a complete discussion of directory entries

The following sections describe directory entries related to Enterprise User Security:

- Enterprise Users
- Enterprise Roles
- Enterprise Domains
- Database Server Entries
- User-Schema Mappings
- Administrative Groups
- Password Policies

1.1.3.1 Enterprise Users

An *enterprise user* is one who is defined and managed in a directory. Each enterprise user has a unique identity across an enterprise. Enterprise user entries can reside at any location within the identity management realm, except within the realm Oracle Context.

> **Note:**
>
> When creating enterprise users in a 9.0.4 or later Oracle Internet Directory, use the tools that come with that 9.0.4 or later Oracle Internet Directory, such as Delegated Administration System (DAS). Even if your databases are 9*i* or 9*i*R2, do not use the 9*i* or 9*i*R2 Enterprise Security Manager GUI tool to create users in a 9.0.4 or later Oracle Internet Directory.
>
> Use only DAS-based tools, like the Oracle Internet Directory Self-Service Console, that ship with Oracle Application Server 10*g*, to create enterprise users in identity management realms.

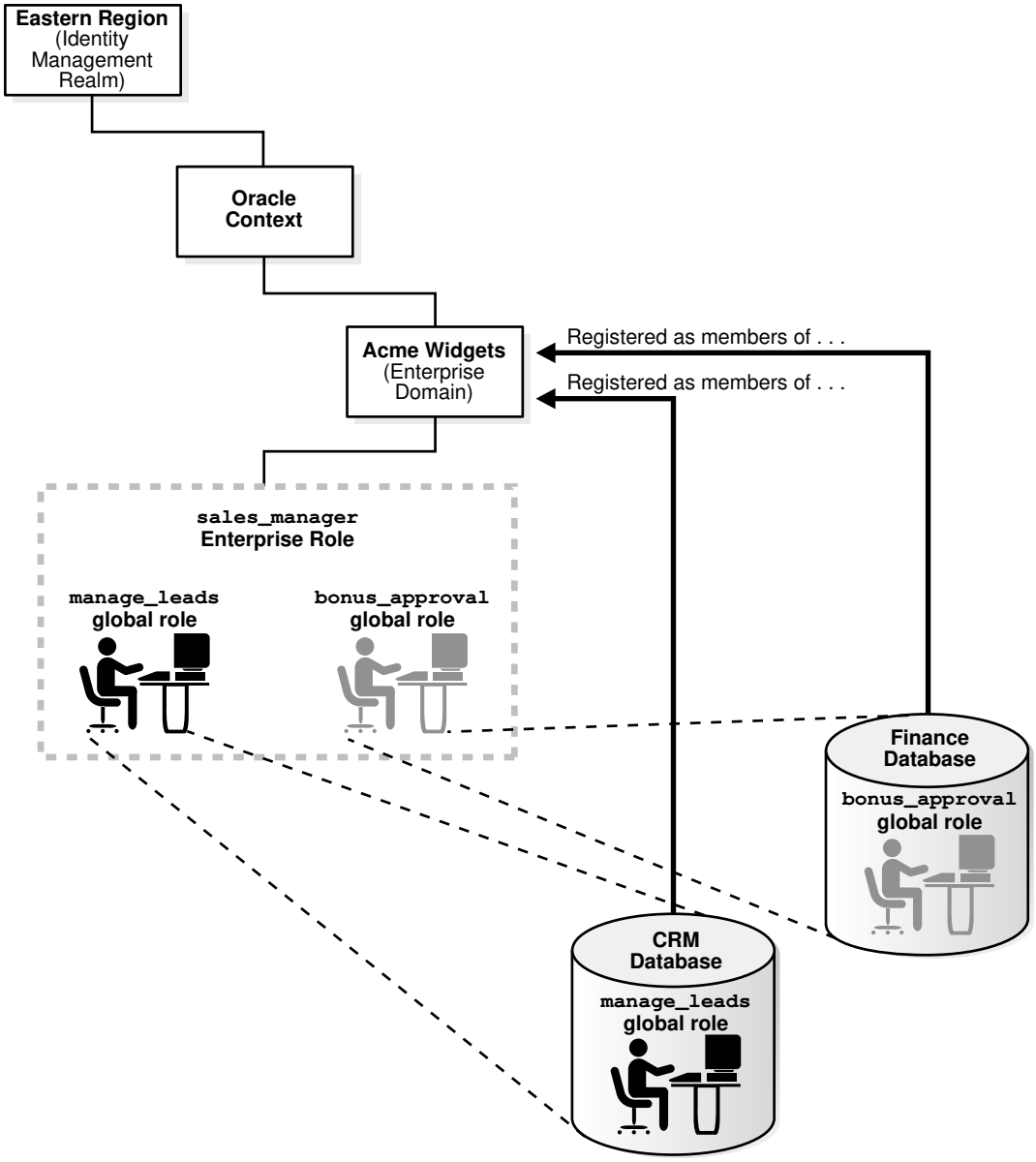
The entries described in the following sections can reside only within a realm Oracle Context.

1.1.3.2 Enterprise Roles

An *enterprise role* is a directory object that acts like a container to hold one or more database global role**s**. Each global role is defined in a specific database where it is assigned privileges, but then it is managed in the directory by using enterprise roles. Enterprise users can be assigned an enterprise role, which determines their access privileges on databases. Figure 1-3 shows an example of an enterprise role called Manager under OracleDefaultDomain.

As an example, consider the enterprise role `sales_manager`, which contains the global role `manage_leads` with its privileges on the Customer Relationship Management (CRM) database, and the `bonus_approval` global role with its privileges on the Finance database. Figure 1-2 illustrates this example.

Figure 1-2 Example of Enterprise Roles



An enterprise role can be assigned to one or more enterprise users. For example, you could assign the enterprise role `sales_manager` to a number of enterprise users who hold the same job. This information is protected in the directory, and only a directory administrator can manage users and assign their roles. A user can be granted local roles and privileges in a database in addition to enterprise roles, by virtue of the privileges on the schema to which the user connects.

Enterprise role entries are stored in enterprise domain subtrees. Each enterprise role contains information about associated global roles on each database server and the associated enterprise users. The enterprise domain administrator creates and manages enterprise roles by using Oracle Enterprise Manager.

> **See Also:**
>
> "Configuring Enterprise Roles" for information about using Oracle Enterprise Manager to create and manage enterprise roles

> **Note:**
>
> The database obtains a user's global roles from the directory as part of the login process. If you change a user's global roles in the directory, then those changes do not take effect until the next time the user logs in to the database.

1.1.3.3 Enterprise Domains

An *enterprise domain* is a group of databases and enterprise roles. An example of a domain could be the engineering division in an enterprise or a small enterprise itself. Figure 1-3 shows an example of an enterprise domain called Services that resides under the OracleDBSecurity entry in an identity management realm. It is here, at the enterprise domain level, that the enterprise domain administrator, using Oracle Enterprise Manager, assigns enterprise roles to users and manages enterprise security.

An enterprise domain subtree in a directory is composed of three types of entries: enterprise role entries, user-schema mappings, and the enterprise domain administrator's group for that domain. Enterprise domains are used to manage information that applies to multiple databases. All user-schema mappings entries contained in an enterprise domain apply to all databases in the domain. If you need to apply different user-schema mappings to individual databases, then use database server entries, which are discussed in the following section.

Enterprise roles apply to specific databases in the domain, as explained in the previous section. Enterprise roles, domain-level mappings, and the domain administrators group are all administered by using Oracle Enterprise Manager.

> **See Also:**
>
> "Administering Enterprise Domains"

1.1.3.4 Database Server Entries

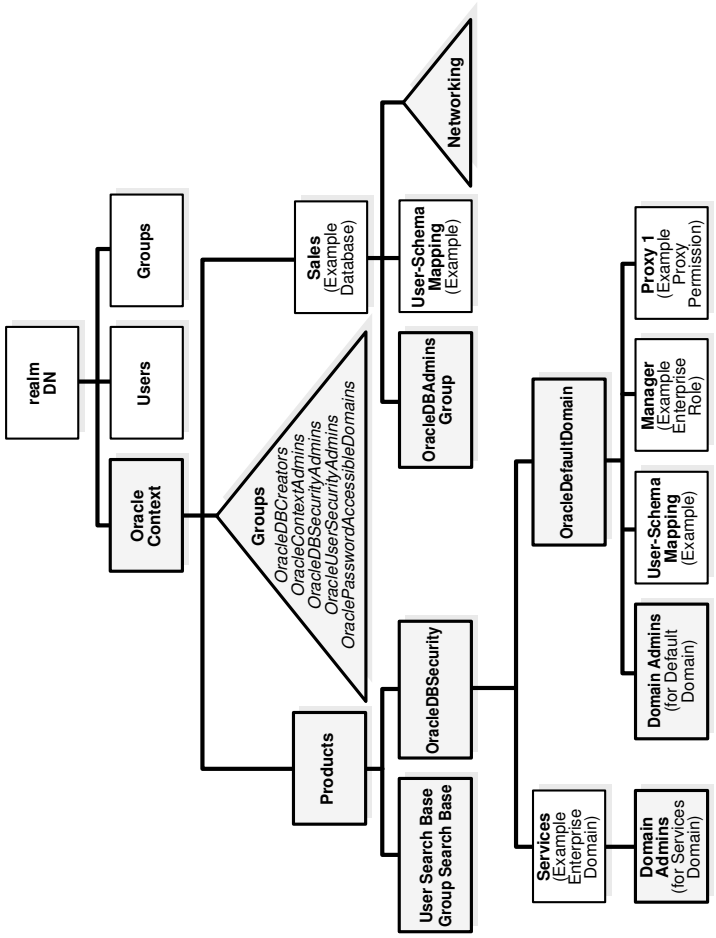
A *database server entry* (represented as "Sales" in Figure 1-3) is a directory entry containing information about one database server. It is created by the Database Configuration Assistant during database registration. A database server entry is the parent of database-level mapping entries called user-schema mappings, which describe mappings between full or partial user DNs and database shared schema names. User-schema mapping entries are created by the database administrator by using Oracle Enterprise Manager.

> **See Also:**
>
> "Oracle Enterprise Manager"

Database administrators belong to the directory administrative group, OracleDBAdmins, which is also managed with Oracle Enterprise Manager. Only OracleDBAdmins or OracleContextAdmins group members can add or remove users from the OracleDBAdmins group. When a user registers a database in the directory, Database Configuration Assistant automatically puts the person who performs registration into the OracleDBAdmins group. The directory entry for this group is located under the database server entry in the DIT.

> **See Also:**
>
> - Table 1-2 for a description of the OracleContextAdmins group
> - "Task 6: Register the database in the directory"
> - "Administering Enterprise Domains" and "Adding Administrators to Manage Database Schema Mappings"

Figure 1-3 Related Entries in a Realm Oracle Context



1.1.3.5 User-Schema Mappings

A *user-schema mapping* is a directory entry that contains mapping information between a user's DN and an Oracle database schema. The users referenced in the mapping are connected to the specified schema when they connect to the database. User-schema mapping entries can apply to only one database or they can apply to all databases in a domain, depending on where they reside in the realm Oracle Context.

See Also:

- "How Enterprise Users Are Mapped to Schemas"
- "Creating User-Schema Mappings for an Enterprise Domain"

1.1.3.6 Administrative Groups

An identity management realm contains administrative groups related to Enterprise User Security. Figure 1-3 shows these administrative groups in a realm in the triangle

labeled "Groups." Each administrative group includes Access Control Lists (ACLs) that control access to the group itself. ACLs elsewhere in the directory may refer to these groups, which allows directory administrators access to perform necessary administrative tasks. The administrative user who creates the realm automatically becomes the first member of each of these groups, thus gaining the associated privileges provided by each group. However, this user can be removed.

The relevant administrative groups in a realm are described in Table 1-2.

> **Note:**
>
> Observe the following practices. Using other methods may break the security configuration for Enterprise User Security objects and may break enterprise user functionality as well.
>
> - Do not modify the ACLs for the objects contained in a realm Oracle Context. Modified realm Oracle Context object ACLs are not supported.
> - Use only Oracle tools, such as Oracle Enterprise Manager, Oracle Internet Directory Self-Service Console, and Database Configuration Assistant, to modify Enterprise User Security directory entries.

Table 1-2 Administrative Groups in a Realm Oracle Context

Administrative Group	Description
OracleContextAdmins	**DN**: (`cn=OracleContextAdmins,cn=Groups,cn=OracleContext...`) **Default owner**: The user who created the identity management realm. (If it is the realm created during installation, then it is `orcladmin`.) OracleContextAdmins has full access to all groups and entries within the associated realm Oracle Context.
OracleDBAdmins	DN: (`cn=OracleDBAdmins,cn=<database_entry_name>,cn=OracleContext...`) Default owner: None. Database Configuration Assistant automatically makes the user who registers a database in the directory a member of this group. Members of this group manage user-schema mapping**s** specific to this database. Only users who are already members of this group or OracleContextAdmins can add or remove users from the OracleDBAdmins group.
OracleDBCreators	**DN**: `(cn=OracleDBCreators,cn=OracleContext...)` **Default owner**: OracleContextAdmins During default realm Oracle Context creation, Oracle Internet Directory Configuration Assistant sets up the following access rights/permissions for these group members: • `Add` permission for database service objects in the realm Oracle Context • `Modify` permission for the Default Domain OracleDBCreators create new databases and register them in the directory by using Database Configuration Assistant

Table 1-2 (Cont.) Administrative Groups in a Realm Oracle Context

Administrative Group	Description
OracleDBSecurityAdmins	**DN**: (cn=OracleDBSecurityAdmins,cn=OracleContext...) **Default owner**: All group members. During default realm Oracle Context creation, Oracle Internet Directory Configuration Assistant sets up the following access rights/permissions for these group members: • All privileges in the OracleDBSecurity subtree • Modify privileges for membership in this group OracleDBSecurityAdmins have permissions on all of the domains in the enterprise and perform the following tasks: • Sets Enterprise User Security configurations for the realm, such as the default database-to-directory authentication method • Group owner administers the OracleDBSecurityAdmins group • Creates and deletes enterprise domains • Moves databases from one domain to another within the enterprise
OracleDomainAdmins	**DN**: (cn=OracleDomainAdmins,cn=<enterprise_domain_name>, cn=OracleDBSecurity,cn=Products,cn=OracleContext....) **Default owner**: The user creating or updating the domain. If a new context and OracleDefaultDomain are created, then the initial member will be the context creator. Members of the OracleDomainAdmins group have full privileges for the enterprise domain. They manage mappings, enterprise roles, and proxy permissions specific to the entire domain. You should be a member of OracleDomainAdmins (for the domain), OracleDBSecurityAdmins, or OracleContextAdmins to modify membership of this group.
OracleUserSecurityAdmins	**DN**: (cn=OracleUserSecurityAdmins,cn=Groups,cn=OracleContext...) **Default owner**: The user who created the identity management realm. By default, an ACL is set at the directory root in Oracle Internet Directory that sets up the relevant permissions so OracleSecurityAdmins can administer Oracle user security.
OraclePasswordAccessibleDomains	**DN**: (cn=OraclePasswordAccessibleDomains,cn=Groups,cn=OracleContext...) **Default owner**: Same as OracleDBSecurityAdmins Group members are enterprise domains, which contain databases enabled for password-authorized enterprise users.

1.1.3.7 Password Policies

Password policies are a set of rules that apply to all user passwords in an identity management realm. Password policies include settings for password complexity, minimum password length, and the like. They also include account lockout and password expiration settings.

A *password policy* entry is defined in Oracle Internet Directory for every identity management realm. Password policies in Oracle Internet Directory are standard Oracle Internet Directory entries that can be used by Oracle Database for Enterprise User Security.

Oracle Internet Directory ensures that all enterprise user passwords meet the rules specified in the password policy entry for the realm. The database communicates with Oracle Internet Directory when authenticating an enterprise user. It requests Oracle Internet Directory to report any password policy violations. If the database gets a policy violation response from Oracle Internet Directory, then it flashes the appropriate warning or error message to the user.

The database reports the following events:

- It flashes a warning when the user password is about to expire and displays the number of days left for the user to change their password.
- It flashes a warning when the password has expired and informs the user about the number of grace logins that remain.
- It displays an error when the user password has expired and the user does not have any grace logins left.
- It displays an error when the user account has been locked due to repeated failed attempts at login.

> **Note:**
>
> For Enterprise SYSDBA users, the failed login count is enabled and is updated whether the database is up or down.

- It displays an error if the user account has been disabled by the administrator.
- It displays an error if the user account is inactive.

Enterprise user login attempts to the database, update the user account status in Oracle Internet Directory. For example, consecutive failed login attempts to the database results in the account getting locked in the directory, as per the directory's password policy.

> **See Also:**
>
> *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for detailed information on password policies and their management

1.2 About Using Shared Schemas for Enterprise User Security

The following sections describe shared schemas, and how to set them up:

- Overview of Shared Schemas Used in Enterprise User Security
- How Shared Schemas Are Configured for Enterprise Users
- How Enterprise Users Are Mapped to Schemas

1.2.1 Overview of Shared Schemas Used in Enterprise User Security

Users do not necessarily require individual accounts or schemas set up in each database. Alternatively, they can connect to a shared schema and be granted access to the objects associated with target applications. For example, suppose that users Tom, Dick, and Harriet

require access to the Payroll application on the Finance database. They do not need to create unique objects in the database, and therefore do not need their own schemas, but they do need access to the objects in the Payroll schema.

Oracle Database supports mapping multiple users stored in an enterprise directory to a shared schema on an individual database. This separation of users from schemas reduces administration costs by reducing the number of user accounts on databases. It means that you do not need to create an account for each user (user schema) in addition to creating the user in the directory. Instead, you can create a user in the enterprise directory, and map that user to a shared schema. Other enterprise users can also be mapped to that schema.

For example, if Tom, Dick and Harriet all access both the Sales and the Finance databases, you do not need to create an account for each user on each database. Instead, you can create a single shared schema on each database, such as `GUEST`, that all three users can access. Then individual access to objects in the Sales or Finance database can be granted to these three users by using enterprise roles. A typical environment can have up to 5,000 enterprise users mapped to one shared schema and each user can be assigned a set of enterprise roles.

Oracle recommends that you create a separate shared schema that contains no objects to use as an entry point. Then, grant access to application objects in other schemas through enterprise roles. Otherwise, application objects can be inadvertently or maliciously deleted or altered.

In summary, shared schemas provide the following benefits:

- Shared schemas eliminate the need to have a dedicated database schema on each database for each enterprise user.
- Each enterprise user can be mapped to a shared schema on each database the user needs to access. The user connects to the shared schema when the user connects to a database.
- Shared schemas lower the cost of managing users in an enterprise.

1.2.2 How Shared Schemas Are Configured for Enterprise Users

To configure shared schemas, the local database administrator (DBA) must create at least one database schema in a database. Enterprise users can be mapped to this schema.

In the following example, the administrator creates a shared schema and maps users to it:

1. The administrator creates a global shared schema called `EMPLOYEE` and the global role `HRMANAGER` on the HR database.
2. The administrator uses the Oracle Internet Directory Self-Service Console and Oracle Enterprise Manager to create and manage enterprise users and roles in the directory. For example, the administrator creates enterprise user Harriet and an enterprise role named `MANAGER`. The administrator then assigns the HR database global role of `HRMANAGER` to the enterprise role `MANAGER`.
3. The administrator assigns enterprise roles to enterprise users in the directory. For example, the administrator assigns the enterprise role `MANAGER` to Harriet.
4. The administrator uses Oracle Enterprise Manager to map the user Harriet in the directory to the shared schema *EMPLOYEE* on the HR database.

When Harriet connects to the HR database, she is automatically connected to the `EMPLOYEE` schema and is given the global role of `HRMANAGER`. Multiple enterprise users can be mapped to the same shared schema. For example, the enterprise security administrator can create another enterprise user Scott and map Scott to the `EMPLOYEE` schema. From that point on, both Harriet and Scott automatically use the `EMPLOYEE` schema when connecting to the HR database, but each can have different roles and can be individually audited.

> **See Also:**
>
> *Oracle Database Security Guide* for more information about auditing

1.2.3 How Enterprise Users Are Mapped to Schemas

Global schemas (those created with `CREATE USER IDENTIFIED GLOBALLY AS ''`) can be owned by one enterprise user (exclusive schema) or shared among multiple enterprise users (shared schema). The mapping between a single enterprise user and their exclusive schema is stored in the database as an association between the user DN and the schema name. The mapping between enterprise users and a shared schema is done in the directory by means of one or more mapping objects. A mapping object is used to map the distinguished name (DN) of a user to a database schema that the user will access. You create a mapping object by using Oracle Enterprise Manager. This mapping can be one of the following:

- Entry-level (full DN) mapping

  This method associates the DN of a single directory user with a particular schema on a database. It results in one mapping entry for each user.

- Subtree-level (partial DN) mapping

  This method lets multiple enterprise users share part of their DN to access the same shared schema. This method is useful if multiple enterprise users are already grouped under some common root in the directory tree. The subtree that these users share can be mapped to a shared schema on a database. For example, you can map all enterprise users in the subtree for the engineering division to one shared schema, BUG_APP_USER, on the bug database. Note that the root of the subtree is not mapped to the specified schema.

  When an enterprise user connects to a database, the database retrieves a DN for the user, either from the network (in the case of SSL) or from the directory (in the case of password and Kerberos-authenticated enterprise users).

  When determining which schema to connect the user to, the database uses the user DN and the following precedence rules:

  1. It looks for an exclusive schema locally (in the database).
  2. If it does not find an exclusive schema locally, then it searches the directory. Within the directory, it looks under the database server entry, first for an entry-level mapping, and then for a subtree-level mapping.
  3. If it does not find a mapping entry under the server entry, then it looks under the enterprise domain entry, first for an entry-level mapping, and then for a subtree-level mapping.

4. If it does not find an exclusive schema locally or an applicable mapping entry in the database, then the database refuses the connection. Otherwise, the database connects the user to the appropriate schema.

For example, suppose that Harriet is trying to connect to the HR database but the database does not find Harriet's exclusive schema (in the database). In this case, the following events occur:

1. The HR database looks up a user schema mapping with Harriet's DN in the directory. The directory has a mapping of Harriet to the shared schema `EMPLOYEE` and returns this schema.
2. The database logs Harriet in and connects her to the `EMPLOYEE` schema.
3. The database retrieves this user's global roles for this database from the directory.
4. The database also retrieves from its own tables any local roles and privileges associated with the database schema to which the user is mapped.
5. The database uses both the global and the local roles to determine the information that the user can access.

Continuing this example, assume that the enterprise role `MANAGER` contains the global roles `ANALYST` on the HR database, and `USER` on the Payroll database. When Harriet, who has the enterprise role `MANAGER`, connects to the HR database, she uses the schema *EMPLOYEE* on that database.

- Her privileges on the HR database are determined by:
  - The global role `ANALYST`
  - Any local roles and privileges associated with the *EMPLOYEE* schema on the HR database
- When Harriet connects to the Payroll database, her privileges are determined by:
  - The global role `USER`
  - Any local roles and privileges associated with the `EMPLOYEE` schema on the Payroll database

You can grant privileges to a specified group of users by granting roles and privileges to a database schema. Every user sharing such a schema gets these local roles and privileges in addition to personal enterprise roles. However, you should exercise caution when doing this because every user who is mapped to this shared schema can exercise the privileges assigned to it. Accordingly, Oracle does not recommend granting roles and privileges to a shared schema.

See Also:

- "Task 1: Create Global Schemas and Global Roles in the Database" for detailed information about how to create shared schemas for enterprise users
- "Enterprise User Proxy"

1.3 Enterprise User Proxy

Sometimes, an enterprise user needs to connect to a database as another user, temporarily having the target user's authorizations and privileges. This capability is particularly useful for midtier tools or applications, which often operate across various databases as enterprise users, their identities established as entries in Oracle Internet Directory. Such an application can maintain a single database connection while switching end user identities, thereby providing functionality in the name of each authorized user in turn.

Enterprise User Security 11*g* Release 1 (11.1) enhanced the efficiency of the proxy mechanism by introducing a single-session model. The two-session proxy model required maintaining separate sessions for the proxy user and the target user. In the new model, only one session is maintained in the security context of the target user. This leads to an improvement in performance.

Enterprise User Security 11*g* Release 1 (11.1), and later, allows greater granularity in assigning proxy permissions to enterprise users. Enterprise users can be individually granted permissions to proxy as local database users. The permissions no longer need to be associated with the user's shared schema in the database.

Being able to assign proxy permissions individually to enterprise users means that the permissions can be more specific. Assigning permissions to a shared schema, on the other hand, forces you to assign the same permissions to all users who map to the schema. This can lead to unwarranted rights and privileges.

Enterprise user proxy permissions are created and stored in Oracle Internet Directory. A permission allows one or more enterprise users or groups to proxy as a target database user. Permissions can apply to specific databases or to all databases in the enterprise domain.

By default, domain administrators can manage proxy permissions in the directory for an enterprise domain. These permissions are configured and managed using Oracle Enterprise Manager.

> **See Also:**
>
> For more information on configuring enterprise user proxy permissions, see "Configuring Proxy Permissions"

Setting up such proxying has several stages:

1. Identify all enterprise users who need permissions to proxy to various databases.
2. Identify all the target users in each such database.
3. Issue `ALTER USER` commands for each such target user, in the following form:

   - `ALTER USER` *target_user* `GRANT CONNECT THROUGH ENTERPRISE USERS`

   The *`target_user`* can now be proxied to by the enterprise users that have proxy permissions in Oracle Internet Directory. Revoking proxy permission uses similar syntax, replacing `GRANT` with `REVOKE`.

> **See Also:**
>
> For the full `ALTER USER` syntax, see *Oracle Database SQL Language Reference*
>
> For Oracle Call Interface usage, see *Oracle Call Interface Programmer's Guide*

4. Grant proxy permissions to each enterprise user either individually or as a member of a group. See the section entitled "Granting Proxy Permissions to Enterprise Users".

> **Note:**
>
> To establish a group representing those enterprise users who will proxy to the same database user, use Oracle Delegated Administration Services as described in the *Oracle Identity Management Guide to Delegated Administration*.

5. With all four of the preceding steps accomplished, your identified enterprise users can proxy to any of the local database users you identified and associated with them. Two versions of the `CONNECT` command can be used. In (a), you supply the enterprise user's password in the command. In (b), you do not, relying instead on the password being in a wallet whose location was put in the sqlnet.ora file.

   a. To establish an enterprise user proxy connection as a database user, use the following SQL*Plus command syntax, supplying the enterprise user's password:

      - ```
        CONNECT joeproxy[targetuser]@database_service_name
        Enter Password:
        ```

        where you would replace *joeproxy* with the name of the enterprise user wishing to proxy as *targetuser*, and replace *targetuser* with the name of the registered user of the target database. The square brackets are required. Enter the enterprise user's password when prompted for the password.

      Once these identities are validated, this connection request results in a single session, in which the proxy user operates in the target database as the target user. The identity of the original user is maintained through to the database, and the audit records can capture both the proxy and the target user's identity.

   b. To connect as an enterprise user proxy for a database user without specifying a password, ensure that the `sqlnet.ora` file contains the location of the wallet holding that user's password. Then, use the following command syntax:

      - ```
        CONNECT  [targetuser]/@database_service_name
        ```

        where you would replace *targetuser* with the name of the registered user of the target database. The square brackets are required. The current enterprise user proxies as the *targetuser*.

> **Note:**
>
> The regular proxy login mechanism using OCI calls can still be used. The `CONNECT` syntax is a new alternative. For more information on the OCI call mechanism, refer to *Oracle Database Security Guide*.

Although the enterprise user proxy permissions are assigned in Oracle Internet Directory, the database administrator can decide which local accounts are to be available as enterprise user proxy targets. The enterprise domain administrator can assign proxy permissions to only those targets that are available in the *dba_proxies* view of the database.

1.4 Enterprise User Security Deployment Considerations

Consider the following issues before deploying Enterprise User Security:

- Security Aspects of Centralizing Security Credentials
- Security of Password-Authenticated Enterprise User Database Login Information
- Considerations for Defining Database Membership in Enterprise Domains
- Choosing Authentication Types between Clients, Databases, and Directories for Enterprise User Security

1.4.1 Security Aspects of Centralizing Security Credentials

Beyond the general benefits that flow from the centralization of enterprise users and their associated credentials, there are a number of security-related benefits and risks that should be reviewed. The following sections describe these benefits and risks:

- Security Benefits Associated with Centralized Security Credential Management
- Security Risks Associated with Centralized Security Credential Management

1.4.1.1 Security Benefits Associated with Centralized Security Credential Management

Centralizing management makes it easier and faster to administer users, credentials, and roles, and to quickly revoke a user's privileges on all applications and databases across the enterprise. With centralized management, the administrator can delete a user in one place to revoke all global privileges, minimizing the risk of retaining unintended privileges.

Centralizing management makes it possible to centralize an organization's security expertise. Specialized, security-aware administrators can manage all aspects of enterprise user security, including directory security, user roles and privileges, and database access. This is a substantial improvement over the traditional model, where DBAs are typically responsible for everything on the databases they manage, including security.

1.4.1.2 Security Risks Associated with Centralized Security Credential Management

While Oracle Internet Directory is a secure repository, there is a security challenge and inherent risk in centralizing credentials in any publicly accessible repository. Although centralized credentials can be protected at least as securely as distributed credentials, the very nature of centralization increases the consequences of inadvertent credential exposure

to unauthorized parties. It is therefore imperative to limit the privileges of administrators to set restrictive Access Control Lists (ACLs) in the directory, and to implement good security practices in the protection of security credentials when they are temporarily outside of the directory.

1.4.2 Security of Password-Authenticated Enterprise User Database Login Information

In all secure password-based authentication methods, a server authenticates a client with a password verifier, typically a hashed version of the password that must be rigorously protected.

Password-based authentication to an Oracle database is no different. There is a password verifier, and it must be protected as well. This is true if the verifier is stored locally in the database or centrally in the directory. Note that a password verifier cannot be used to derive the original password.

An enterprise user's database password can be stored in a central directory service for access by multiple databases. It can be viewed and shared by all trusted databases to which the user has access. Although the password verifier stored in the directory is not the cleartext password, it is still necessary to protect it from casual or unauthorized access. It is therefore extremely important to define password-related ACLs in the directory that are as restrictive as possible while still enabling necessary access and usability.

Oracle tools help set up ACLs in the directory to protect these password verifiers during identity management realm creation. The approach that Oracle recommends is intended to balance security and usability considerations. If you require maximum security and can set up wallets for all users, you should require only SSL connections from users to databases. This SSL-only approach circumvents the entire directory password protection issue.

The following sections provide more information about trusted databases and protecting database password verifiers in the directory:

- What Is Meant by Trusted Databases
- Protecting Database Password Verifiers

1.4.2.1 What Is Meant by Trusted Databases

SSL provides strong authentication so databases are ensured of being trusted with their own identity. With password-authenticated Enterprise User Security where database password verifiers are stored centrally in a directory and shared among multiple databases, each database that allows password-authenticated enterprise users to log in must be a trusted database. Each database has access to the shared password verifiers, so it is important that each database can be trusted to observe the following security precautions:

- Each database must be trusted to protect itself from tampering with the server code so a malicious user cannot misuse the database identity to gain access to password verifiers in the directory.
- Each database must be trusted to protect its PKI and other credentials from theft so a malicious user cannot use them to gain access to the password verifiers stored in the directory.

1.4.2.2 Protecting Database Password Verifiers

The OraclePasswordAccessibleDomains group in each identity management realm is created automatically when the realm is created, and it can be managed by using Oracle Internet Directory tools like the Oracle Internet Directory Self-Service Console. Enterprise domains with member databases that must view users' database password verifiers in the directory are placed in this group.

For a selected realm, determine which databases can accept password-authenticated connections. Use Oracle Internet Directory Self-Service Console to place the domains containing those databases into the OraclePasswordAccessibleDomains group. An ACL on the user subtree permits access to the directory attribute that holds the password verifier used by the database.

All other users are denied access to this attribute. An ACL that prevents anonymous read access to the password verifier attributes is at the root of the directory tree.

Note that for usability, by default, OracleDefaultDomain is a member of the OraclePasswordAccessibleDomains group. It can be removed, if desired.

1.4.3 Considerations for Defining Database Membership in Enterprise Domains

Consider the following criteria when defining the database membership of a domain:

- Accepted authentication types for enterprise users are defined at the domain level. Database membership in a domain should therefore be defined accordingly.

  > **Note:**
  >
  > If one or more databases are intended to only support SSL-based certificate authentication, they cannot be combined in the same domain with password-authenticated databases.

- Enterprise roles are defined at the domain level. To share an enterprise role across multiple databases, the databases must be members of the same domain.

1.4.4 Choosing Authentication Types between Clients, Databases, and Directories for Enterprise User Security

Enterprise User Security supports the authentication types listed in Table 1-3 for connections between clients, databases, and directories.

Table 1-3 Enterprise User Security: Supported Authentication Types for Connections between Clients, Databases, and Directories

Connection	Supported Authentication Types
Clients-to-Databases	Passwords, SSL, and Kerberos
Databases-to-Directories	SSL and Passwords

However, some combinations of authentication types for connections make more sense than others. For example, it is unusual to have a high level of security for client-database connections by using SSL for all user connections, but then configuring the database to authenticate to the directory by using passwords. Although this configuration is supported, it does not provide consistent security for connections. Ideally, the database-directory connection should be at least as secure as that between users and databases.

The following section describes some typical configurations: Typical Configurations.

1.4.4.1 Typical Configurations

The following combinations of authentication types between clients, databases, and directories are typical:

- Password authentication for all connections
- SSL authentication for all connections
- Kerberos authentication for client-to-database connections, and password authentication for database-to-directory connections

2
Getting Started with Enterprise User Security

Enterprise User Security enables you to centrally manage database users across the enterprise. Enterprise users are created in Oracle Internet Directory, and can be assigned roles and privileges across various enterprise databases registered with the directory.

This chapter uses a tutorial approach to help you get started with Enterprise User Security. The following steps discuss configuring Enterprise User Security:

1. Configuring Your Database to Use the Directory
2. Registering Your Database with the Directory
3. Creating a Shared Schema in the Database
4. Mapping Enterprise Users to the Shared Schema
5. Connecting to the Database as an Enterprise User
6. Using Enterprise Roles
7. Using Proxy Permissions
8. Using Pluggable Databases

2.1 Configuring Your Database to Use the Directory

The first step in configuring Enterprise User Security is to configure the database to use the directory. Running the Net Configuration Assistant (NetCA) tool enables you to configure the directory host name and port that your database should use.

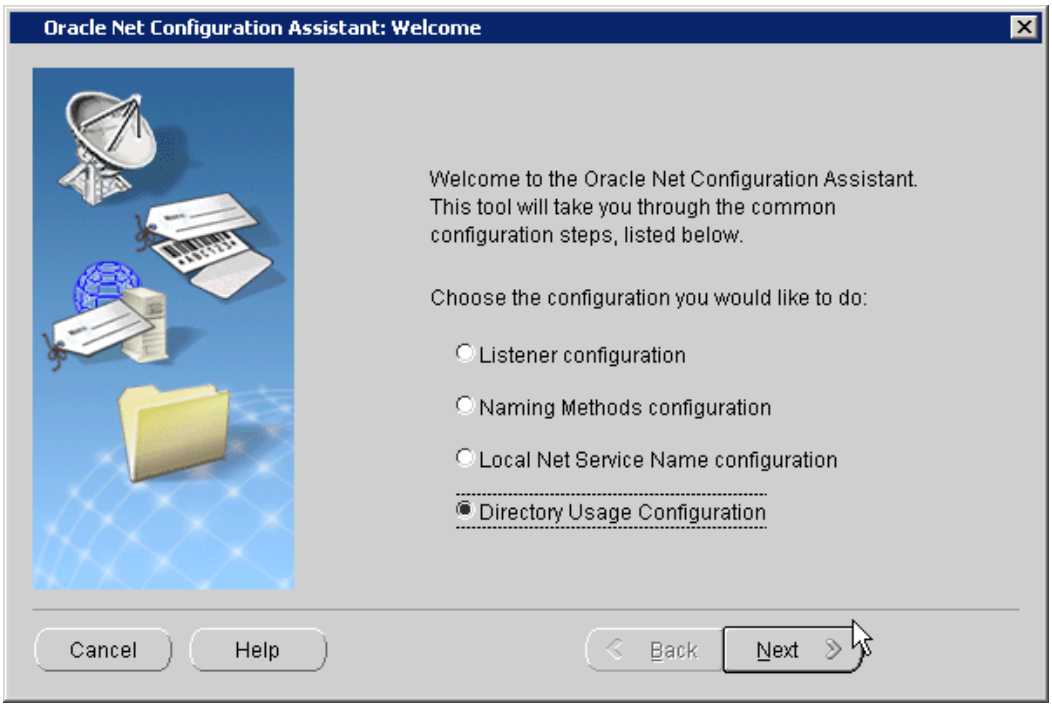
To configure your database for directory usage:

1. Start NetCA using the `netca` command.
   - On Windows, you can also start NetCA from the Start menu:

     Click **Start, All Programs, Oracle - *OracleHomeName***, **Configuration and Migration Tools, Net Configuration Assistant.**
   - On Unix systems, you can start NetCA using the following command:

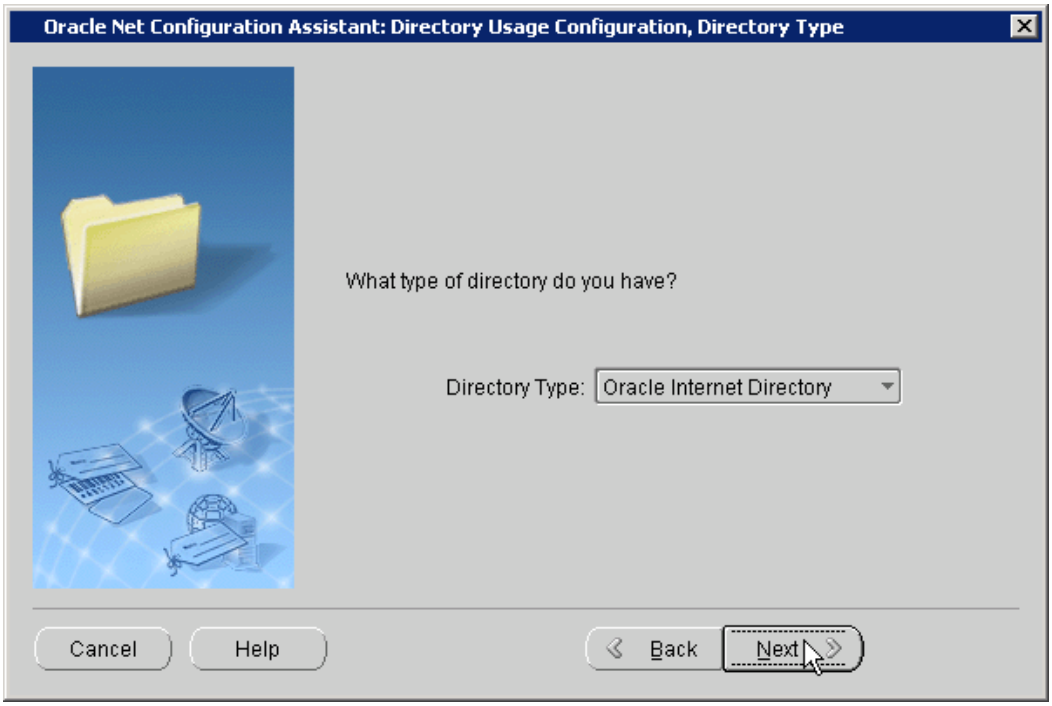
     $ORACLE_HOME`/bin/netca`

   The Welcome screen appears.



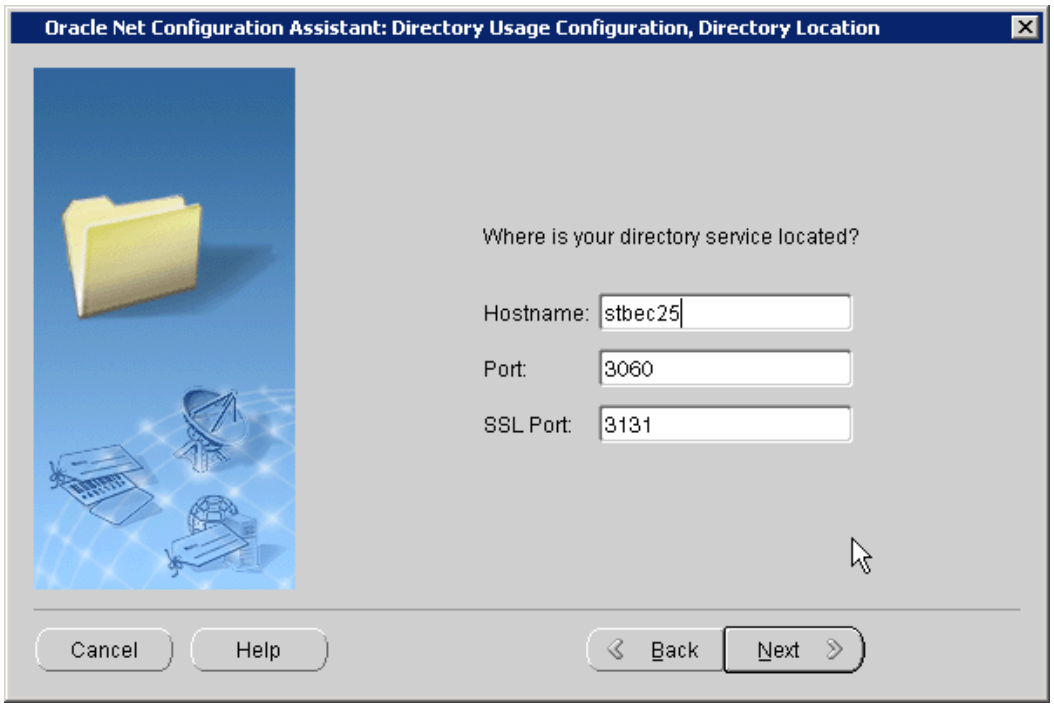
2. Select **Directory Usage Configuration**. Click **Next**.

   The Directory Type screen appears.



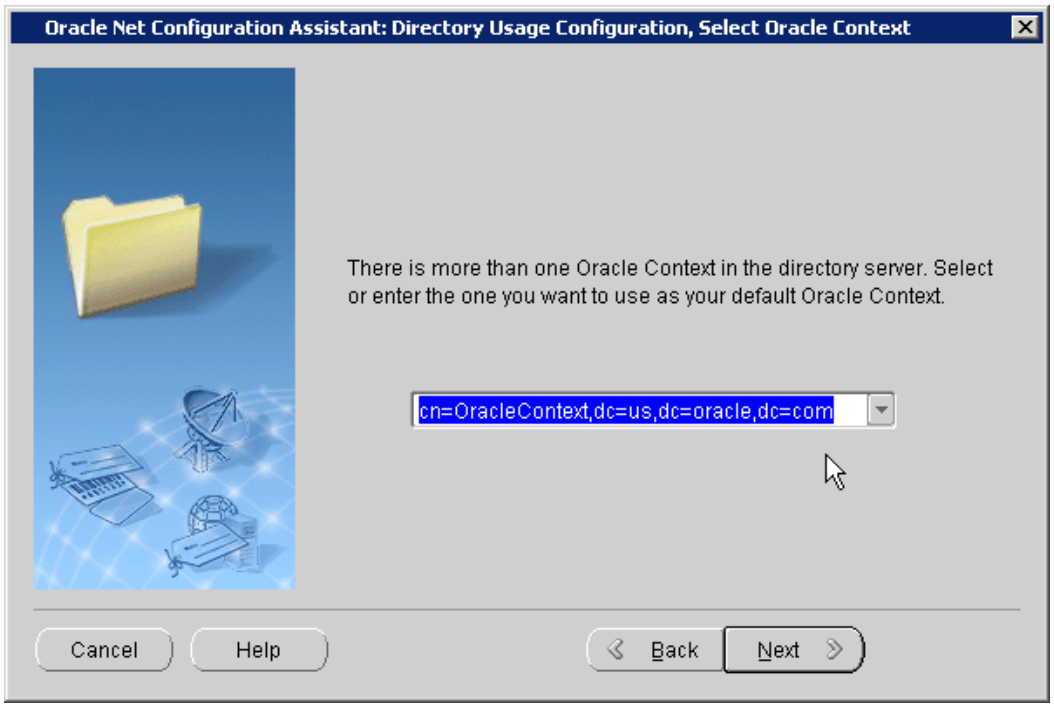
3. Click **Next**.

   The Directory Location screen appears.



4. Enter the name of the host on which the Oracle Internet Directory server is running. Also enter the LDAP non-SSL and SSL port numbers. These port numbers are 3060 and 3131 by default. Click Next.

   The Select Oracle Context screen appears.



5. Select the default Oracle Context to use. You need to select this if there are multiple identity management realms on the directory server. Click **Next**.

   The Directory Usage Configuration, Done screen is displayed.

6. Confirm that the directory usage configuration is successfully completed. Click **Next**.
7. Click **Finish**.

   NetCA creates an `ldap.ora` file in the *`$ORACLE_HOME`*`/network/admin` directory. This is the *`$ORACLE_HOME`*`\network\admin` directory in Windows. The `ldap.ora` file stores the connection information details about the directory.

2.2 Registering Your Database with the Directory

The next step is to register the database with the directory service. The Database Configuration Assistant (DBCA) tool enables you to register the database with Oracle Internet Directory.

To register the database with the directory:

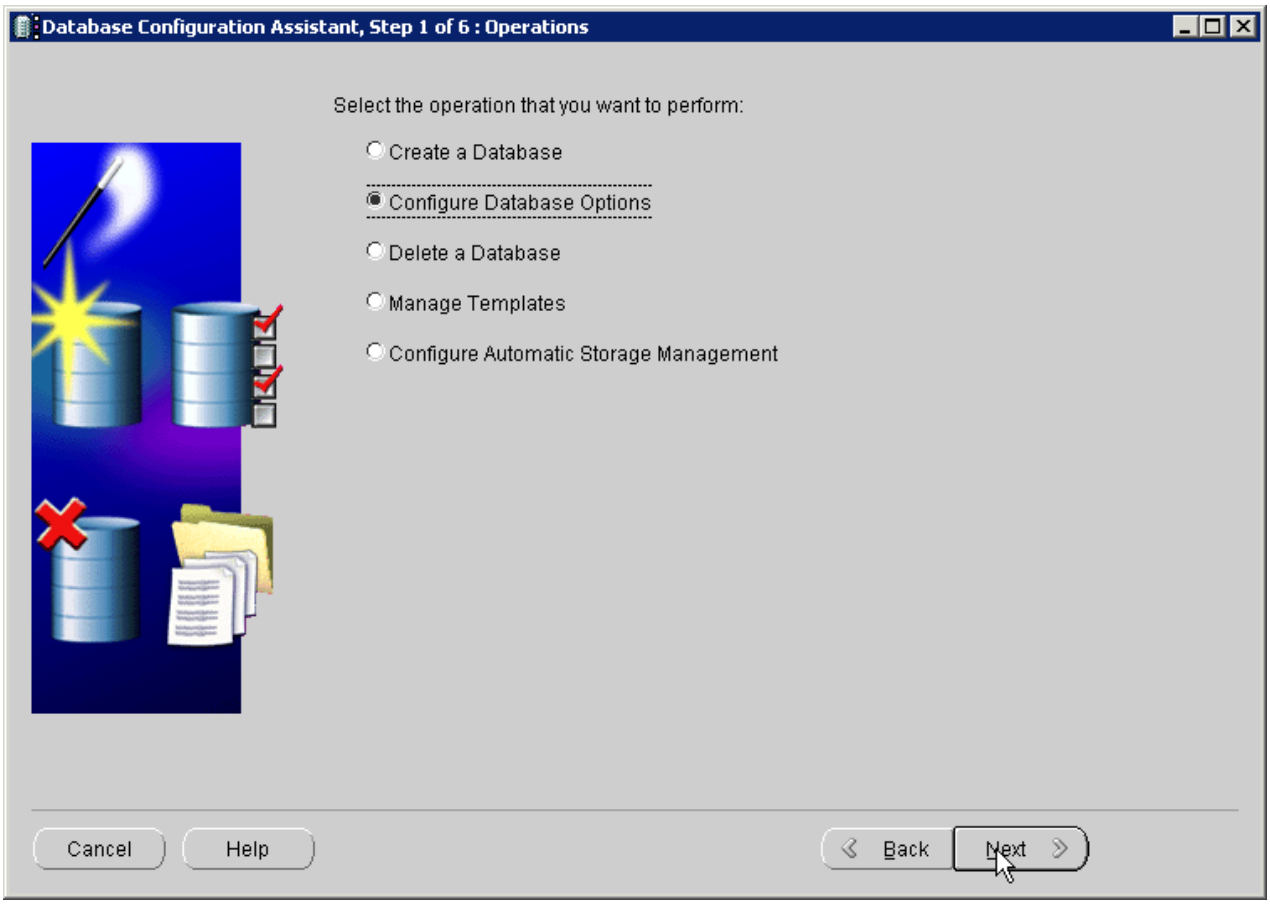
1. Start DBCA using the `dbca` command.
   - On Windows, you can also start DBCA from the Start menu:

     Click **Start, All Programs, Oracle - *OracleHomeName***, **Configuration and Migration Tools, Database Configuration Assistant.**
   - On Unix systems, you can start DBCA using the following command:

     `$ORACLE_HOME``/bin/dbca`

   The Welcome screen appears.
2. Click **Next**.

   The Operations screen displays.



3. Select **Configure Database Options**. Click **Next**.

   The Database screen appears.

4. Select the database name that you wish to configure. You might also be asked to enter `SYS` user credentials if you are not using operating system authentication. Click **Next**.

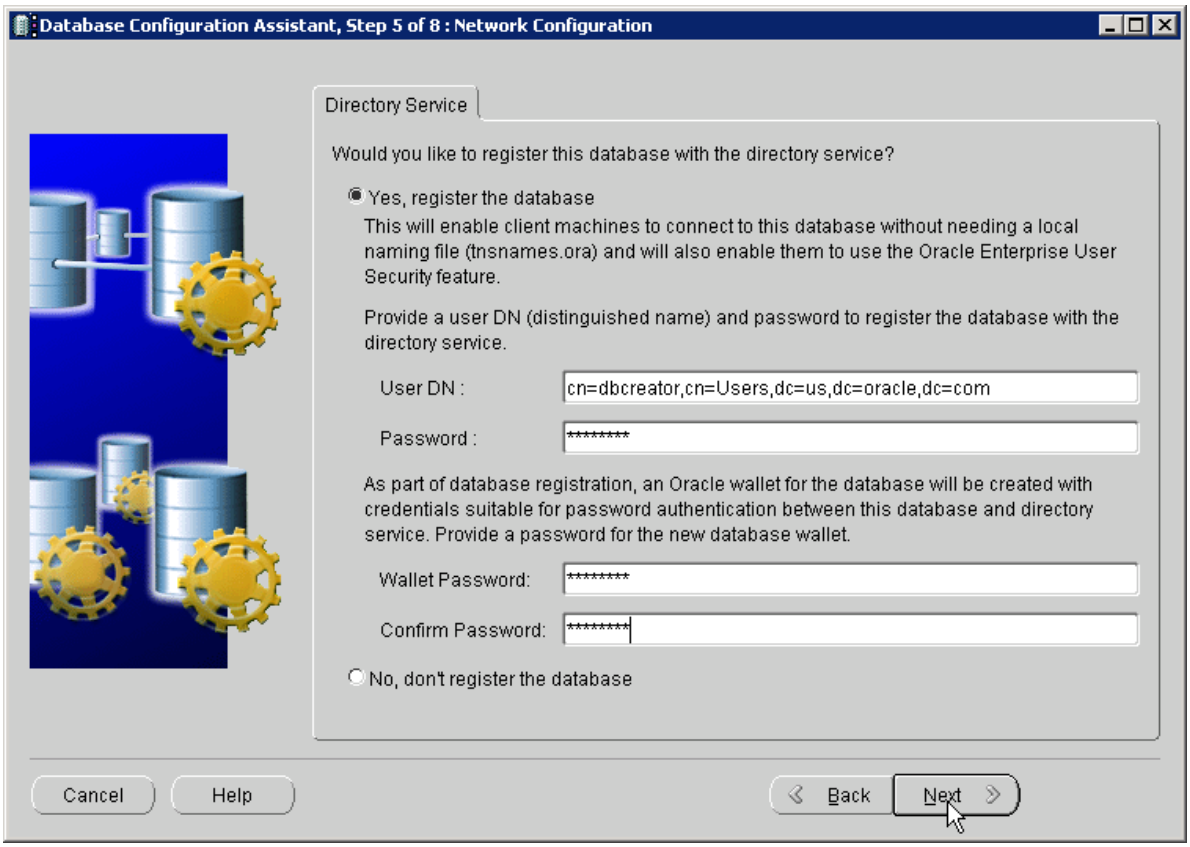
   The Management Options screen appears.

5. Select the management options, and click **Next**.

   The Security Settings screen appears.

6. Select **Keep the enhanced 12c default security settings** to keep the Oracle Database 12*c* security settings. Click **Next**.

   The Network Configuration screen appears.



7. Select **Yes, register the database** to register the database with the directory. Enter the distinguished name (DN) of a user who is authorized to register databases in Oracle Internet Directory. Also, enter the password for the directory user. Enter a wallet password. Reenter the password in the **Confirm Password** field. Click **Next**.

> **Note:**
>
> The database uses a randomly generated password to log in to the directory. This database password is stored in an Oracle wallet. The wallet can also be used to store certificates needed for SSL connections.
>
> The wallet password that you specify is different from the database password. The wallet password is used to protect the wallet.

   The **Database Components** screen appears.

8. Click **Next**.

   The **Connection Mode** page appears.

9. Select **Dedicated Server Mode** or **Shared Server Mode**. Click **Finish**.

   The **Confirmation** dialog box appears.

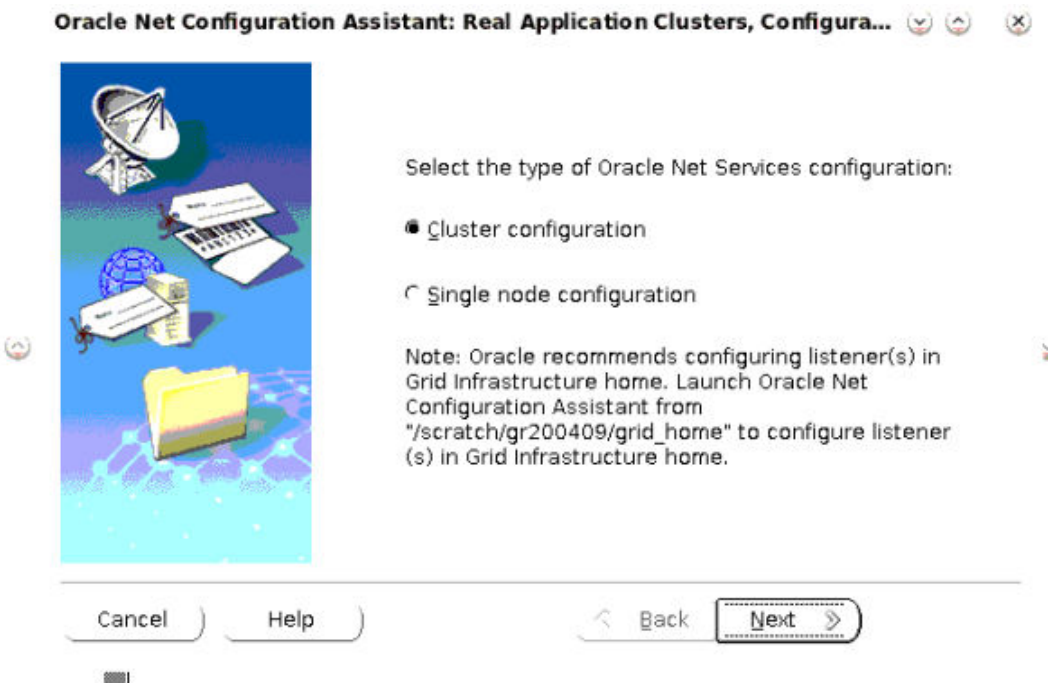
10. Click **OK**.

> **Note:**
>
> After you register the database with the directory, make sure that auto login is enabled for the database wallet. The default wallet is created in the *$ORACLE_BASE*/admin/*database_sid*/wallet directory.
>
> You can verify that auto login for the wallet is enabled by checking for the presence of the cwallet.sso file in the wallet directory.

2.3 Registering an Oracle RAC Database with the Directory

Configuring the Oracle Internet Directory (OID) in an Oracle RAC environment requires certain additional steps.

1. Create the ldap.ora file using NetCA following the steps under Configuring Your Database to Use the Directory. Choose **Cluster configuration** for the entire cluster configuration or **Single node configuration** to create the ldap.ora file in one node and copy it to the other remaining nodes. The ldap.ora file is located at $ORACLE_HOME/network/admin/ldap.ora.



2. Configure the database by running DBCA in silent mode for Oracle RAC databases:

```
dbca -silent -configureDatabase -sourceDB <DB_name> -
registerWithDirService <true/false>
-dirServiceUserName <OracleContext_value> -dirServicePassword <OID_pwd> -
walletPassword <wallet_pwd>
```

DB_name is the name of the source database to configure

`registerWithDirService` is the boolean value that must be set to `true` for configuring the database

`OracleContext_value` is the default Oracle Context that is unique for the OID

`OID_pwd` is the unique password value for the OID

`wallet_pwd` is the user specified wallet password

For example, for an Oracle RAC database named `t2`, the DBCA command is as follows:

```
dbca -silent -configureDatabase -sourceDB t2 -
registerWithDirService true
-dirServiceUserName cn=dbcauser,cn=users,dc=us,dc=oracle,dc=com
-dirServicePassword <OID_pwd> -walletPassword <wallet_pwd>
```

2.4 Creating a Shared Schema in the Database

Creating a shared schema in the database enables you to map multiple enterprise users to the same schema. Example 2-1creates a shared schema, `global_ident_schema_user`, and grants the `CONNECT` role to it.

Example 2-1 Creating a Shared Schema

```
SQL> CREATE USER global_ident_schema_user IDENTIFIED GLOBALLY;
User created.
SQL> GRANT CONNECT TO global_ident_schema_user;
Grant succeeded.
```

2.5 Mapping Enterprise Users to the Shared Schema

Enterprise User Security can be managed using Enterprise Manager. Example 2-2 maps the DN, `cn=users, dc=us, dc=oracle, dc=com` to the shared database schema, `global_ident_schema_user`.

Example 2-2 Mapping Enterprise Users to the Shared Schema

To create the user-schema mapping:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

> **See Also:**
>
> Registering Your Database with the Directory, specifically Step 6, where you previously entered the user DN and password to register the database with the directory service

The Enterprise User Security page appears.

6. Click **Manage Enterprise Domains**.

   The Manage Enterprise Domains page appears.

7. Select the enterprise domain which contains the database. Click **Configure**.

   The Configure Domain page appears.

8. Click the **User-Schema Mappings** tab. All user-schema maps that apply to the enterprise domain are displayed.

9. Click **Create**.

   The Create Mapping page is displayed.

10. Under the From section, select **Subtree**. Click the Search icon. Select the DN, **cn=Users, dc=us,dc=oracle,dc=com**.

11. Under the To section, enter global_ident_schema_user in the **Schema** field. Click **Continue**.

    The user-schema mapping is added in the Configure Domain page.

12. Click **OK**.

2.6 Connecting to the Database as an Enterprise User

All users in the mapped Oracle Internet Directory subtree can now connect to the database as enterprise users. Example 2-3 shows the `cn=orcladmin, cn=users, dc=us,dc=oracle,dc=com` user connecting to the database.

Example 2-3 Connecting to the Database as an Enterprise User

```
SQL> CONNECT orcladmin
Enter password:
Connected.
```

2.7 Using Enterprise Roles

Enterprise roles are created in the directory. Enterprise roles contain global roles from different databases that are part of the enterprise domain. Enterprise roles are used to assign database privileges to enterprise users.

Example 2-4 creates two enterprise users, Joe and Nina. Both these users are created in the subtree, `cn=Users, dc=us,dc=oracle,dc=com`, which is already mapped to the `global_ident_schema_user` in the EUSDB database. See RFC4519 — Lightweight Directory Access Protocol (LDAP) : Schema for User Applications for information about the LDAP attribute types shown in the previous subtree example.

Nina is an HR manager. She needs the `SELECT` privilege on the `hr.employees` table in the EUSDB database. Example 2-4 achieves this using enterprise roles.

Example 2-4 Using Enterprise Roles

We start by creating two enterprise users, Joe and Nina. You can create enterprise users using the Oracle Internet Directory Self Service Console.

To create enterprise users, Joe and Nina:

1. Connect to the Oracle Internet Directory Self Service Console. Use the following URL:

   `http://`*`hostname`*`:`*`port`*`/oiddas/`

   Here, *`hostname`* is the name of the host that is running the Oracle Internet Directory server. The *`port`* number is the TCP port number on which the Oracle Internet Directory Self Service Console is running. This is 7777 by default.

2. Click the **Directory** tab.

   The Sign In page appears.

3. Log in as the user that can create users in Oracle Internet Directory.

   The User page appears.

4. Click **Create**.

   The Create User page appears.

5. Enter joe under **User Name**. Enter values for the other required fields. Select Enabled under **Is Enabled**.
6. Click **Submit**.
7. Click **Create Another User**.

   The Create User page appears.
8. Enter Nina under **User Name**. Enter values for the other required fields. Select Enabled under **Is Enabled**.
9. Click **Submit**. Click **OK**.

Next, we create a global role in the database that allows access to the `hr.employees` table. The following SQL*Plus statements create a global role, `hr_access` and grant the necessary privilege to it.

```
SQL> CREATE ROLE hr_access IDENTIFIED GLOBALLY;
Role created.
SQL> GRANT SELECT ON hr.employees TO hr_access;
Grant succeeded.
```

Next, we create an enterprise role called `hr_access` and assign the global role to it. We then assign this enterprise role to the enterprise user, Nina. The enterprise role can be created using Enterprise Manager.

To create the enterprise role, `hr_access`:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.

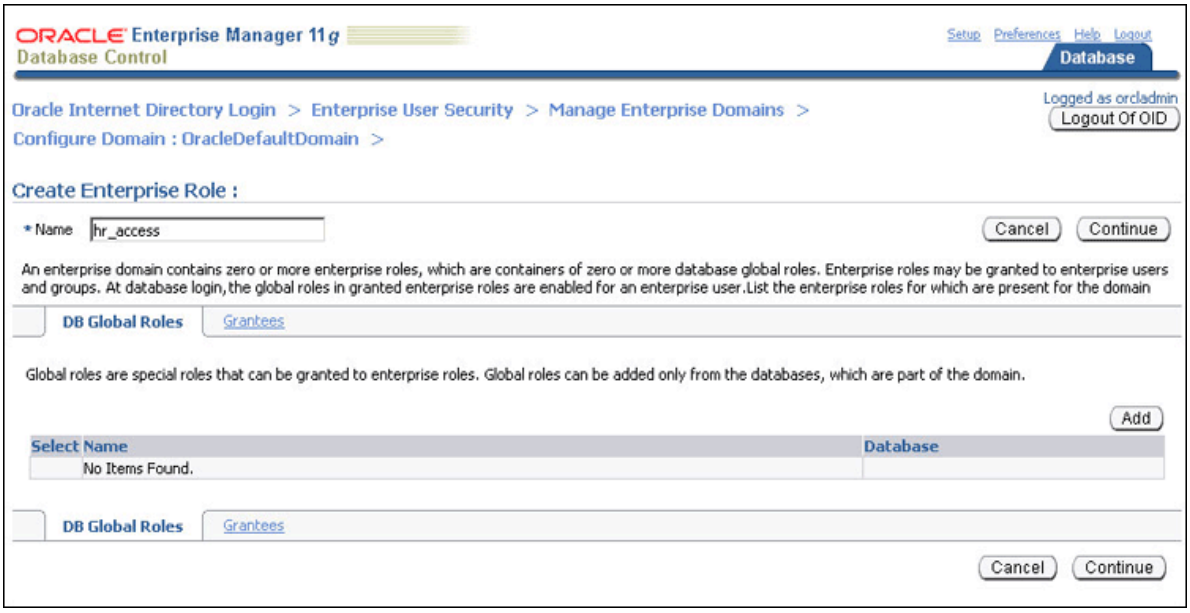
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Manage Enterprise Domains**.

   The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.
7. Select the enterprise domain that contains the database. Click **Configure**.

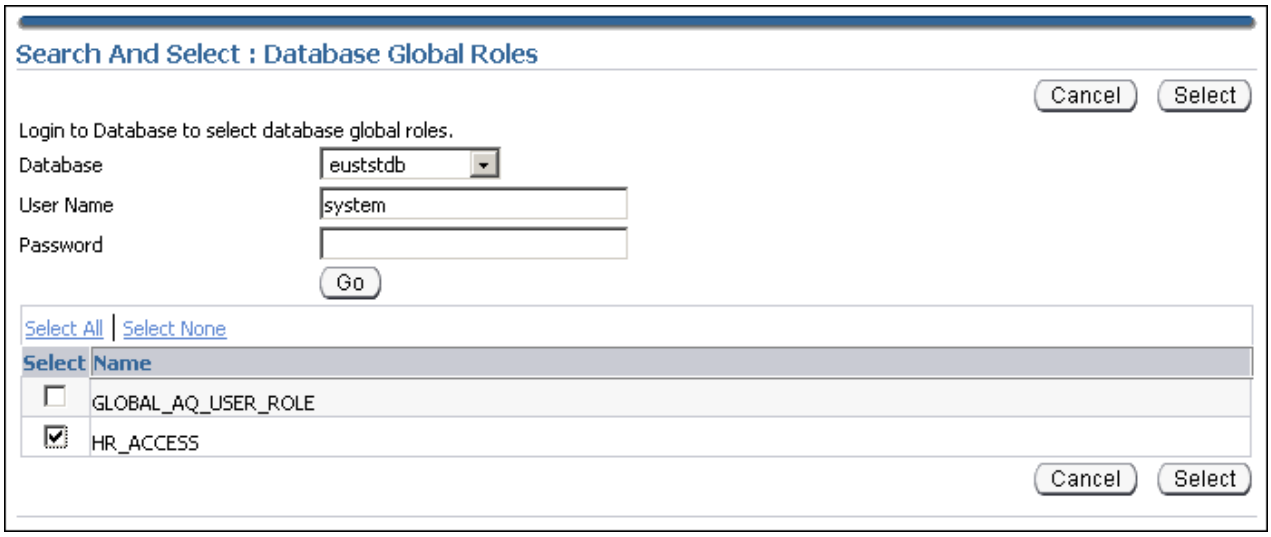
   The Configure Domain page appears.
8. Click the **Enterprise Roles** tab.
9. Click **Create**.

   The **Create Enterprise Role** page appears.
10. Enter hr_access in the **Name** field.



11. Click **Add** to add the database global role to the enterprise role.

    The Search and Select Database Global Roles window is displayed.



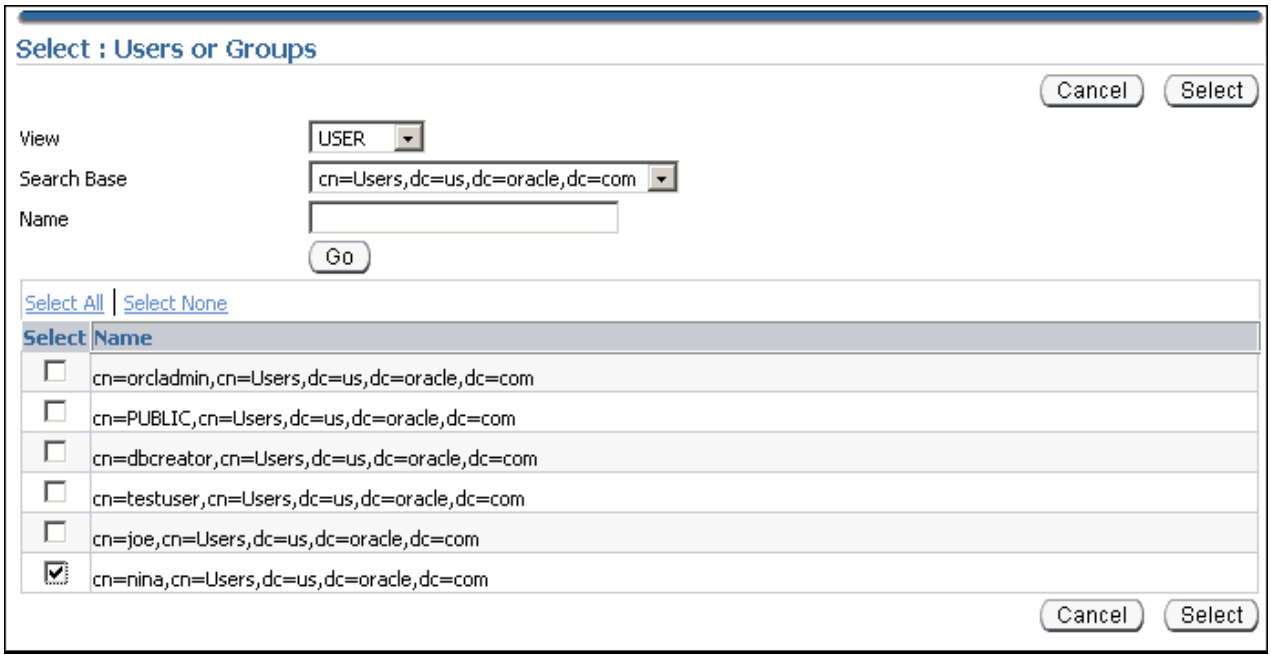
12. Select the `hr_access` global role in your database. Click **Select**.

> **Note:**
>
> You will be required to log in to the database before you can select the global role.

13. Click the **Grantees** tab. Click **Add**.

    The Select Users or Groups window appears.

14. Select user Nina. Click **Select**.



15. Click **Continue** in the Create Enterprise Role page.

16. Click **OK** in the Configure Domain page.

The enterprise user, Nina can now access the `hr.employees` table in the database. The following SQL*Plus statements illustrate this:

```
SQL> CONNECT Nina
Enter password:
Connected.
SQL> SELECT employee_id FROM hr.employees;
EMPLOYEE_ID
-----------
        100
        101
        102
...
...
107 rows selected.
```

The enterprise user, Joe cannot access the `hr.employees` table, as he does not have the enterprise role assigned to him.

```
SQL> CONNECT joe
Enter password:
Connected.
SQL> SELECT employee_id FROM hr.employees;
SELECT employee_id FROM hr.employees

ERROR at line 1:
ORA-00942: table or view does not exist
```

2.8 Using Proxy Permissions

Proxy permissions are created at the enterprise domain level. Proxy permissions allow an enterprise user to proxy a local database user, which means that the enterprise user can log in to the database as the local database user. You can grant proxy permissions to individual enterprise users or groups. Proxy permissions are especially useful for middle-tier applications that operate across multiple databases as enterprise users.

Example 2-5 illustrates the use of proxy permissions. The enterprise user, `joe` is a sales manager and needs to log in to enterprise databases as the target database user, `SH`. The `SH` user owns the sample `SH` schema that contains Sales History related tables.

Example 2-5 Using Proxy Permissions

The first step in allowing enterprise user proxy is to `ALTER` the target database user to allow `CONNECT` through enterprise users. The following `SQL` statements unlock the `SH` database account, set a password for it, and `ALTER` the account to allow enterprise user proxy:

```
SQL> CONNECT SYSTEM
Enter password:
Connected.
SQL> ALTER USER SH IDENTIFIED BY hrd2guess ACCOUNT UNLOCK;
User altered.
SQL> ALTER USER SH GRANT CONNECT THROUGH ENTERPRISE USERS;
User altered.
```

Next, use Enterprise Manager to configure the proxy permission. This allows the enterprise user `joe` to connect as the local database user, `SH`.

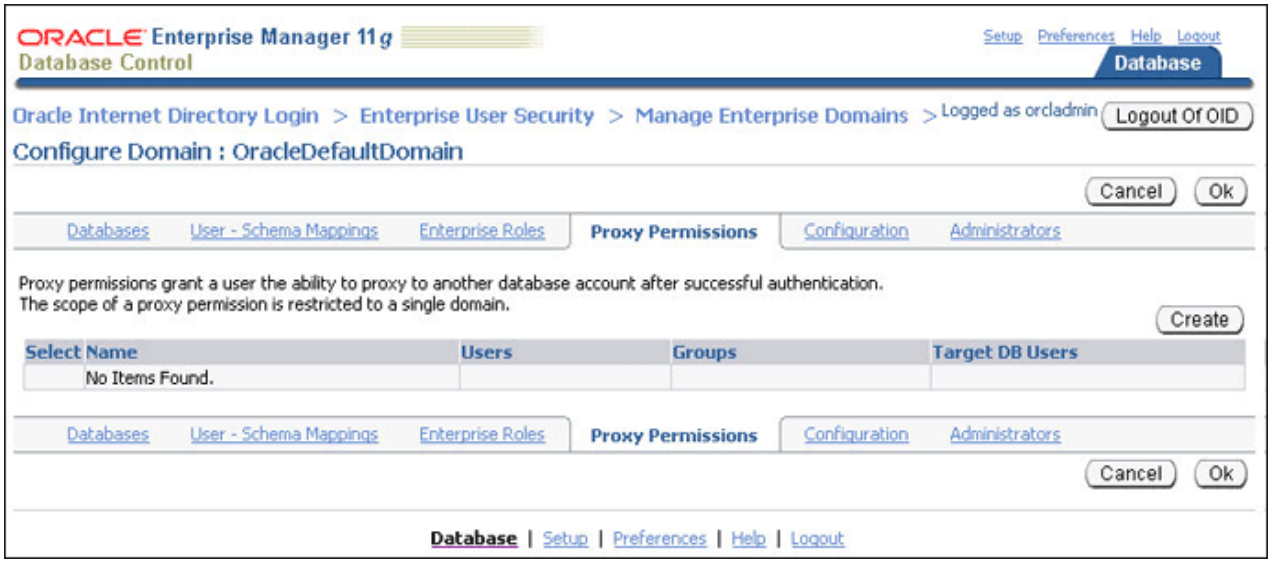
To configure the proxy permission for enterprise user, `joe`:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Manage Enterprise Domains**.

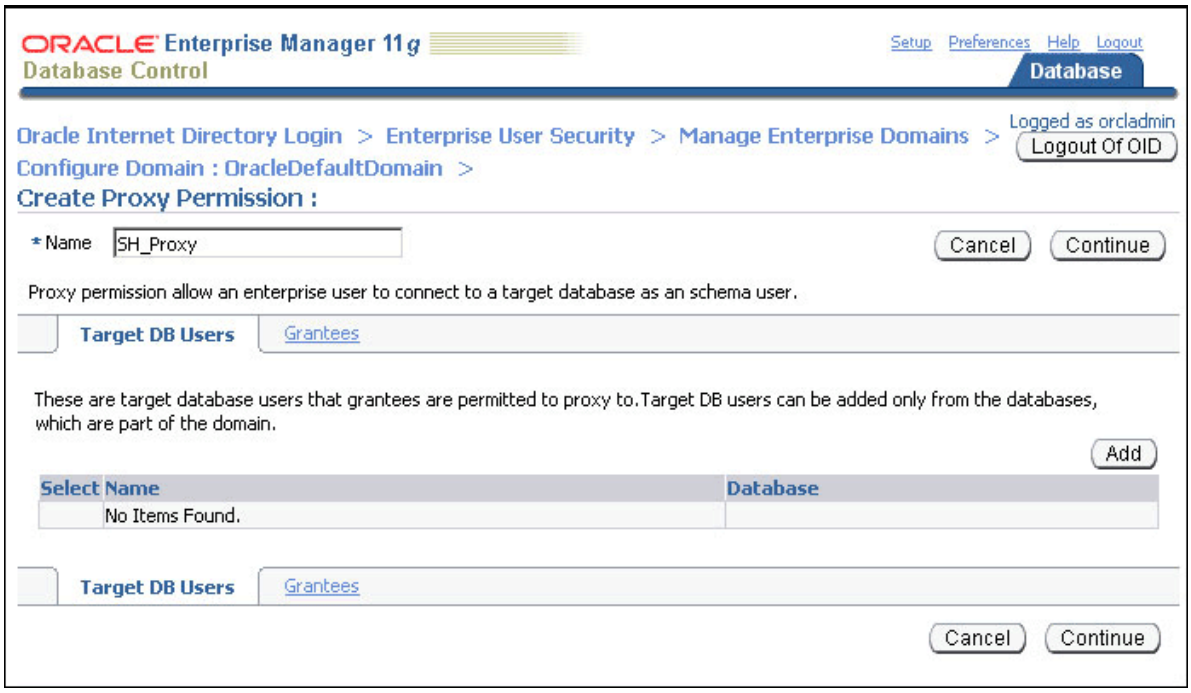
   The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.
7. Select the enterprise domain that you wish to configure. Click **Configure**.

   The Configure Domain page appears.
8. Click the **Proxy Permissions** tab.



9. Click **Create** to create a new proxy permission.

   The **Create Proxy Permission** page appears.
10. Enter SH_Proxy, as the name of the proxy permission, in the **Name** field.



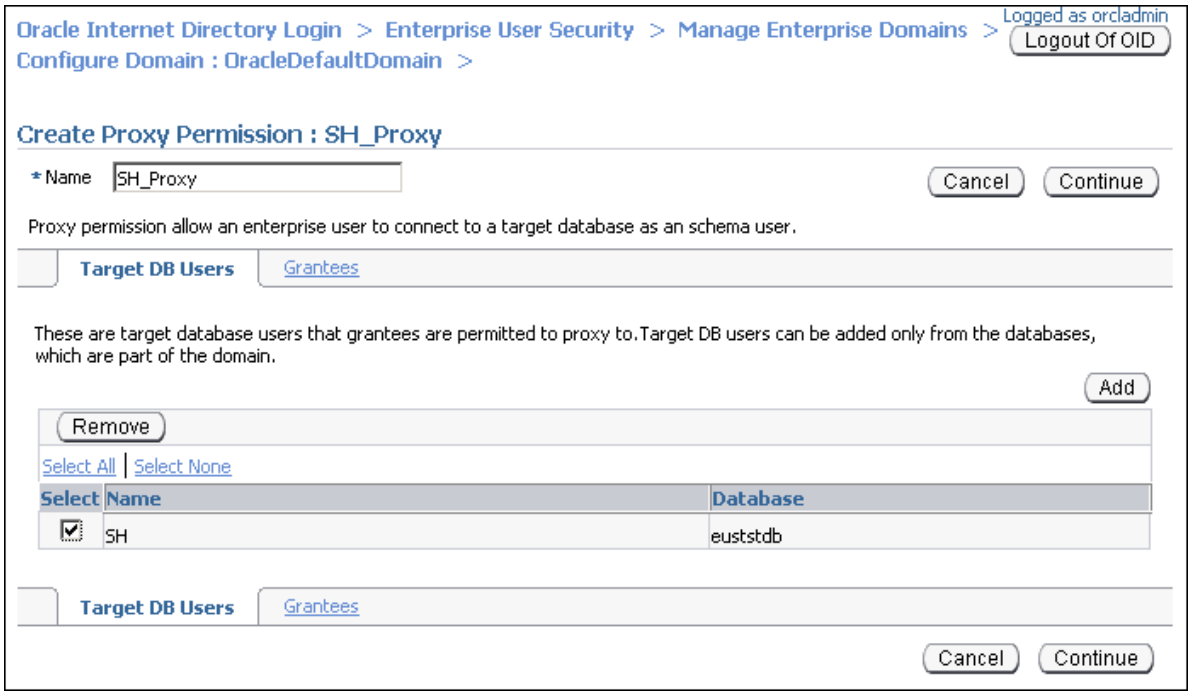
11. Ensure that the **Target DB Users** tab is selected. Click **Add**.

    The Search and Select window appears.

12. Log in to the database that contains the `SH` user. A list of all database users that have been altered to allow enterprise user proxy is displayed.

13. Select the SH user. Click **Select**.

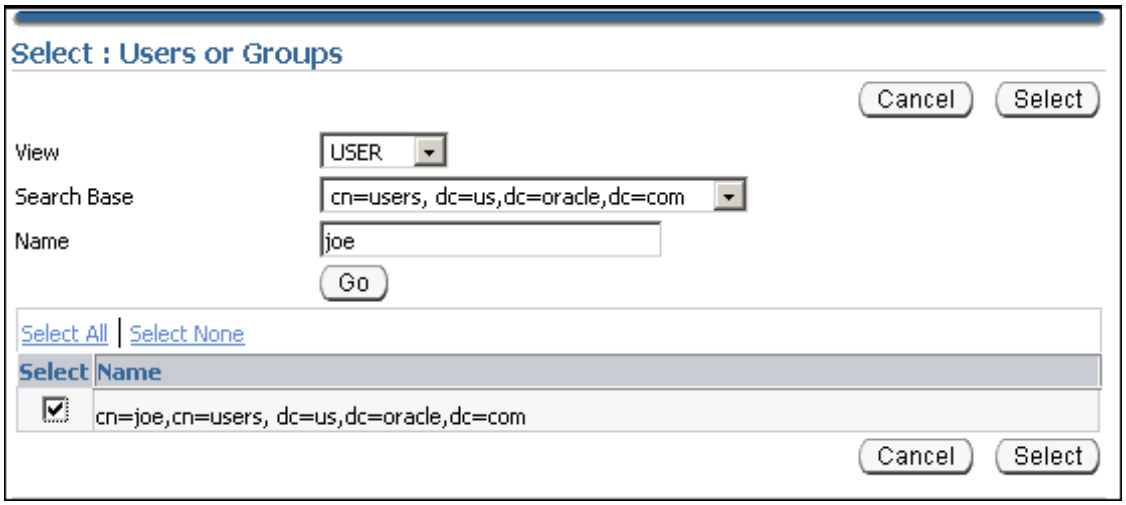
    The `SH` user is added under Target DB Users in the Create Proxy Permission page.



14. Click the **Grantees** tab.
15. Click **Add**.

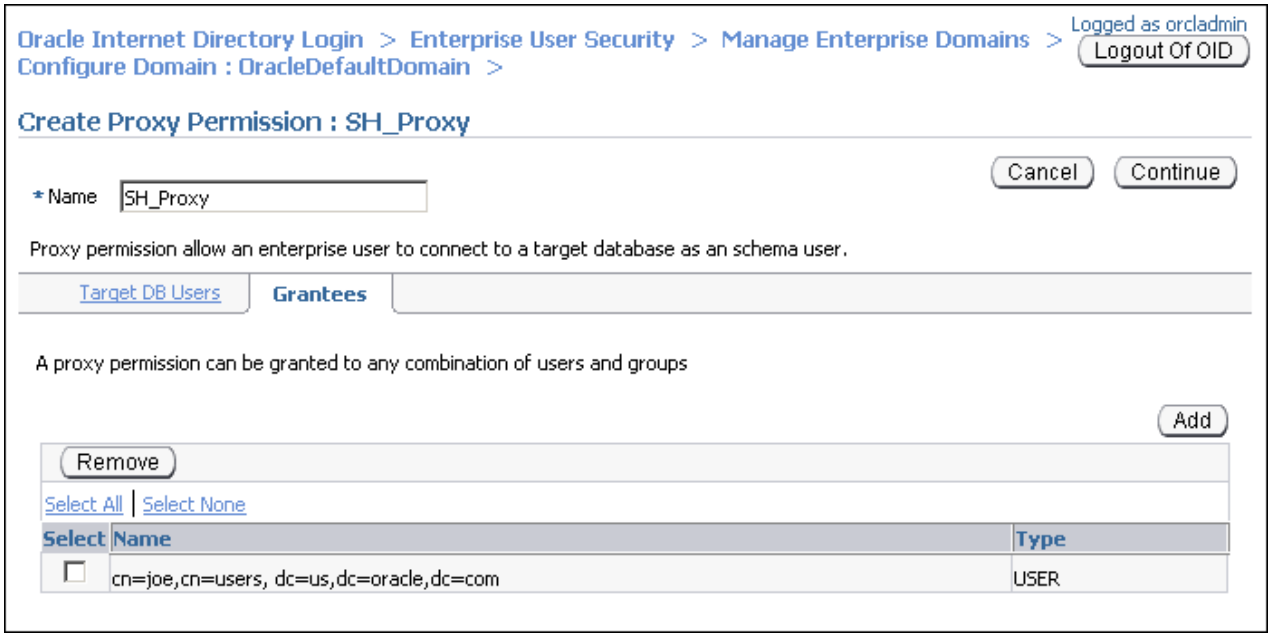
    The Select Users or Groups window appears.
16. Select `cn=users,dc=us,dc=oracle,dc=com` under **Search Base**. Select `User` under **View**. Click **Go**.

    A list of users under the subtree, `cn=users,dc=us,dc=oracle,dc=com` is displayed.



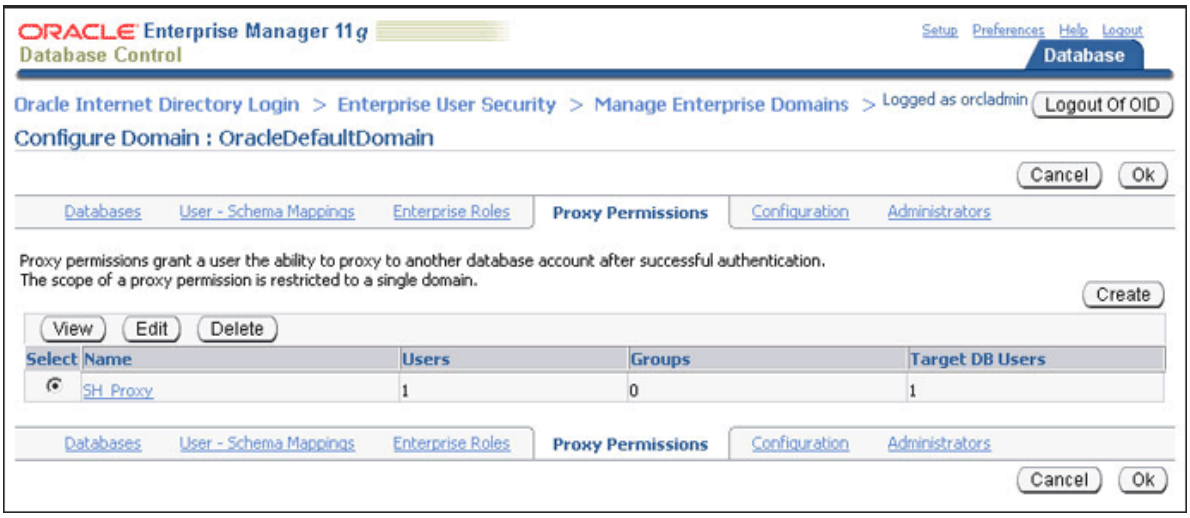
17. Select `cn=joe,cn=users,dc=us,dc=oracle,dc=com`. Click **Select**.

    The user `joe` is added under **Grantees** in the Create Proxy Permission page.



18. Click **Continue** in the Create Proxy Permission page.

    The proxy permission, `SH_Proxy` is added in the Configure Domain page.



19. Click **OK**.

The enterprise user, `joe` can now log in as the local database user `SH`. The following `SQL` statements illustrate this:

```
SQL> REMARK Joe uses his own password to connect as the local database user, SH.
SQL> CONNECT joe[SH]
Enter password:
Connected.
SQL> SELECT * FROM SH.sales WHERE cust_id=4;

   PROD_ID    CUST_ID TIME_ID   CHANNEL_ID   PROMO_ID QUANTITY_SOLD AMOUNT_SOLD
---------- ---------- --------- ---------- ---------- ------------- -----------
        37          4 31-MAY-00          3        999             1       60.43
        39          4 31-MAY-00          3        999             1       38.45
        40          4 31-MAY-00          3        999             1        48.1
...
...
72 rows selected.
```

2.9 Using Pluggable Databases

You can use Enterprise User Security with Pluggable Databases (PDBs), introduced in Oracle Database 12*c* Release 1 (12.1). Each PDB has its own Enterprise User Security metadata, such as global users, global roles, and so on. Each PDB should have its own identity in the directory. A PDB is like any regular database registered with the directory, except for the following restrictions:

- Client-side SSL authentication uses the Container Database (CDB)-wide wallet configured for the listener. The PDB-specific wallet is used for database-to-directory authentication.
- If the client-to-database authentication uses `SSL`, and the database-to-directory authentication also uses `SSL`, then two wallets need to be configured for the database with certificates. The first wallet is the CDB-wide wallet and the second wallet is the PDB-specific wallet.

> **Note:**
>
> Each PDB has it's own value for `LDAP_DIRECTORY_ACCESS` parameter. The value of the `LDAP_DIRECTORY_ACCESS` parameter has to be set from a PDB.

The following sections describe more information about using pluggable databases:

- Wallet Location for Pluggable Databases
- Wallet Root for Pluggable Databases
- Connecting to a Directory Service
- Default Database DN Format
- Plugging and Unplugging PDBs
- Switching Containers

2.9.1 Wallet Location for Pluggable Databases

For pluggable databases, when a PDB is registered with the directory, the Database Configuration Assistant (DBCA) creates the wallet at the following location:

- If the `ORACLE_BASE` environment variable is set:

  `ORACLE_BASE``/admin/`*`db_unique_name`*`/`*`pdb_GUID`*`/wallet`

- If `ORACLE_BASE` is not set:

  `ORACLE_HOME``/admin/`*`db_unique_name`*`/`*`pdb_GUID`*`/wallet`

  The GUID of the PDB is used because the PDB name can change, but the GUID does not change. So, the PDB wallet location is still valid even if the PDB name changes.

> **Note:**
>
> On the Microsoft Windows x64 platform, without specifying the `WALLET_LOCATION` parameter in the `listener.ora` file and server side `sqlnet.ora` file, the server does not pick up the wallets from the default system location, which is *`%USERPROFILE%`*`\ORACLE\WALLETS`. Hence, when you try to login using an SSL connection, the login fails with the following error: `ORA-28864: SSL connection closed gracefully`. There is no workaround to this known issue.

2.9.2 Wallet Root for Pluggable Databases

`WALLET_ROOT` specifies the path to the root of a directory tree containing a subdirectory for each pluggable database (PDB). The directory structure is similar to the Oracle ASM wallet storage directory structure which is used to store the various wallets associated with the PDB.

This example uses the value of `ORACLE_BASE` environment variable to set the root of the wallet directory hierarchy:

```
WALLET_ROOT=$ORACLE_BASE/admin/orcl/wallet
```

You can also use `WALLET_LOCATION` to specify the wallet location of your choice. When you specify both `WALLET_ROOT` and `WALLET_LOCATION` parameters, `WALLET_ROOT` takes the highest precedence. If both parameters are not specified, default wallet location is used.

> **Note:**
>
> This parameter is available starting with Oracle Database release 18c, version 18.1.

> **See Also:**
>
> WALLET_ROOT

2.9.3 Connecting to a Directory Service

Previously in a multitenant database, all the containers used to connect to a single directory service. You can now connect a different directory service for each pluggable database (PDB).

2.9.3.1 Comparison of the dsi.ora and ldap.ora Files

You can use either `dsi.ora` or `ldap.ora` for Enterprise User Security and integration with Oracle Internet Directory(OID).

If you use `ldap.ora` for Enterprise User Security, you can only specify a single OID server for all the containers (PDBs) in a database. However, `ldap.ora` can also be used for net naming services which may or may not be using the same directory service. The `dsi.ora` allows each PDB to integrate with a different OID server if needed. While DBCA can be used to configure the `ldap.ora`, the `dsi.ora` needs to be created manually.

2.9.3.2 About Using a dsi.ora File

You use a `dsi.ora` file to specify the Oracle Internet Directory(OID) server.

You must manually create the `dsi.ora` file to identify the OID server. The `dsi.ora` file provides directory connection information for all pluggable databases if it is located in the same places where the `ldap.ora` file can be placed. A `dsi.ora` file in a PDB-specific wallet location takes precedence over the main `dsi.ora` file for that PDB only.

> **Note:**
>
> If you are using `ldap.ora` for naming services, then do not make any changes to `ldap.ora` for the directory configuration. Only use `dsi.ora` to configure OID.

Placement of dsi.ora

Oracle recommends that you use directories for writable files under `$ORACLE_BASE`, not under `$ORACLE_HOME`. Starting with Oracle Database 18c, you can optionally set the `$ORACLE_HOME` directory to be read-only. Hence, you should place the `dsi.ora` file in a directory that is outside of `$ORACLE_HOME` to accommodate the `dsi.ora` configuration for future releases.

Search Order for dsi.ora

When you create the `dsi.ora` file, Oracle Database searches for it in the following order:

1. For a PDB, Oracle searches for it in the PDB-specific wallet location.
   a. If `WALLET_ROOT` is specified in `init.ora`, the PDB-specific wallet location is at *`WALLET_ROOT`*`/`*`pdb_guid`*`/eus/`
   b. If `WALLET_ROOT` is not specified in `init.ora`, and if `WALLET_LOCATION` is specified in `sqlnet.ora`, the PDB-specific wallet location is at *`WALLET_LOCATION`*`/`*`pdb_guid`*`/`
   c. If neither `WALLET_ROOT` or `WALLET_LOCATION` is specified, the PDB-specific wallet location is the default location, that is:
      i. If the `ORACLE_BASE` environment variable is set, the default location is:

         `ORACLE_BASE``/admin/`*`db_unique_name`*`/`*`pdb_guid`*`/wallet/`

      ii. If `ORACLE_BASE` is not set, the default location is:

         `ORACLE_HOME``/admin/`*`db_unique_name`*`/`*`pdb_guid`*`/wallet/`

2. If Oracle Database cannot find `dsi.ora` in the wallet location, then Oracle Database searches for it in the following order. These are the same locations that Oracle Database searches for the `ldap.ora` file.
   a. `$LDAP_ADMIN` environment variable setting
   b. `$ORACLE_HOME/ldap/admin` directory
   c. `$TNS_ADMIN` environment variable setting
   d. `$ORACLE_HOME/network/admin` directory

When to Use dsi.ora

Oracle recommends that you use only `dsi.ora` to connect to a different OID service for each PDB . If both `dsi.ora` and `ldap.ora` are configured in the same database and are both located in the same directory, then `dsi.ora` takes precedence over the `ldap.ora` file. If they are in different directories, then Oracle uses the first one that it finds in the location precedence list above to find the directory server.

The default wallet location for a PDB is the `$ORACLE_BASE/admin/`*`db_unique_name`*`/`*`pdb_guid`*`/wallet/` directory.

To find the *`db_unique_name`*, connect to the CDB root and execute the following query:

```
SELECT DB_UNIQUE_NAME FROM V$DATABASE;
```

To find the *`pdb_guid`*, from the CDB root, execute the following query:

```
SELECT PDB_NAME,GUID FROM DBA_PDBS;
```

2.9.3.3 Creating the dsi.ora File

The `dsi.ora` configuration file sets the information to find the directory server.

To use the `dsi.ora` configuration file:

1. Log in to the host where the Oracle database is located.
2. Locate the wallet location directory to place the `dsi.ora` file. See Search Order for dsi.ora for more information on wallet location. Create the directory for the wallet location if it does not exist. Go to the wallet location directory to create the `dsi.ora` file.
3. Add the following parameters to the `dsi.ora` file:
   - `DSI_DIRECTORY_SERVERS`, which sets the Directory server host and port number, and alternate directory servers. The directory server name must be a fully qualified name. For example:

     ```
     DSI_DIRECTORY_SERVERS =  (ldap-server.examplecorp.com:389:636,
     raffles.examplecorp.com:389:636)
     ```

   - `DSI_DEFAULT_ADMIN_CONTEXT`, specifies the default directory for the creation, modification, or search of the connect identifiers. For example:

     ```
     DSI_DEFAULT_ADMIN_CONTEXT = "o=OracleSoftware,c=US"
     ```

   - `DSI_DIRECTORY_SERVER_TYPE`, which determines the directory server access. You must set it to `OID` for Oracle Internet Directory. Enter this value in upper case.

     ```
     DSI_DIRECTORY_SERVER_TYPE = OID
     ```

2.9.3.4 About Using an ldap.ora File

You can use an `ldap.ora` file to specify the directory server.

If you are already using an `ldap.ora` file for another purpose such as net naming services, then you must use the `dsi.ora` file to configure the directory for user authentication and authorization. Even if the database currently is not using `ldap.ora` for another service, Oracle recommends using `dsi.ora` in case `ldap.ora` will be used at a future time for net naming services.

If `ldap.ora` is being used for naming services, then do not make any changes to `ldap.ora`. Only use `dsi.ora` to configure the directory.

Benefit of Using ldap.ora

The benefit of using `ldap.ora` is that you can use the DBCA graphical interface or the DBCA silent mode to complete configuring the connection to the directory servers. When using `dsi.ora`, the steps to complete configuring the connection to the directory must be done separately.

Once you configure a PDB using DBCA with a `ldap.ora`, you can create an individual `dsi.ora` in the PDB-specific wallet location for that PDB. Move the contents of `ldap.ora` into the corresponding `dsi.ora` with `DSI_` prefix added to each parameter in `dsi.ora`.

Later, you can update `ldap.ora` with different OID servers. After you update `ldap.ora`, use DBCA to configure another PDB, and move the contents of `ldap.ora` into the `dsi.ora` in the PDB-specific wallet location for that PDB. Repeat the process to configure every PDB so that each PDB can connect to different OID servers.

Placement of ldap.ora

Typically, the `ldap.ora` file is stored in the `$ORACLE_HOME/network/admin` directory.

Search Order for ldap.ora

After you create the `ldap.ora` file, Oracle Database searches for it in the following order:

1. `$LDAP_ADMIN` environment variable setting
2. `$ORACLE_HOME/ldap/admin` directory
3. `$TNS_ADMIN` environment variable setting
4. `$ORACLE_HOME/network/admin` directory

Changing the Contents of ldap.ora

If you change the contents of `ldap.ora` after database has been started, then you must either restart the database instance or re-execute the following DDL to make the updated content in `ldap.ora` effective:

```
ALTER SYSTEM SET LDAP_DIRECTORY_ACCESS = 'PASSWORD';
```

2.9.3.5 Creating the ldap.ora File

These steps assume that `ldap.ora` is not being used for net naming services and can be used to set up the connection with the directory.

1. Log in to the host where the Oracle database is located.
2. Choose a directory where to use the `ldap.ora` file, based on the search order for the `ldap.ora` file. (See Related Topics.) If this directory does not exist, then create the directory. Then go to this directory to create the `ldap.ora` file.
3. If the `ldap.ora` file does not exist, then create it by using a text editor.

   If the `ldap.ora` file does exist, create a backup of this file, and then open `ldap.ora`.
4. Add the following parameters to the `ldap.ora` file:
   - `DSI_DIRECTORY_SERVERS`, which sets the Directory server host and port number, and alternate directory servers. The directory server name must be a fully qualified name. For example:

     ```
     DIRECTORY_SERVERS = (ldap-server.examplecorp.com:389:636,
     raffles.examplecorp.com:389:636)
     ```

- `DEFAULT_ADMIN_CONTEXT`, specifies the default directory for the creation, modification, or search of the connect identifiers. For example:

  ```
  DEFAULT_ADMIN_CONTEXT = "o=OracleSoftware,c=US"
  ```

- `DIRECTORY_SERVER_TYPE`, which determines the LDAP server access. You must set it to `OID` for Enterprise User Security. Enter this value in upper case.

  ```
  DIRECTORY_SERVER_TYPE = OID
  ```

2.9.4 Default Database DN Format

When a PDB is registered with the directory using DBCA, the default PDB Distinguished Name (DN) is generated in the following format:

`cn=`*`PDB_NAME.DB_UNIQUE_NAME`*`,cn=oraclecontext,`*`realm`*

You can change the default `cn` (*PDB_NAME.DB_UNIQUE_NAME)* to a custom value in the DBCA registration screen. It cannot be altered after registration.

2.9.5 Plugging and Unplugging PDBs

You can plug existing registered databases into a CDB. You do not need to register the PDB again, as long as you perform the following steps:

- Pick the wallet files from the source location and put them in the new default wallet location for the PDB.
- Set the `LDAP_DIRECTORY_ACCESS` parameter to the desired value in the PDB.

Similarly, when unplugging an existing PDB that is registered with the directory, you do not need to re-register the database. You need to copy the wallet to the new default location after unplug. The default location should be:

- If the `ORACLE_BASE` environment variable is set:

  `ORACLE_BASE``/admin/`*`db_unique_name`*`/`*`pdb_guid`*`/wallet`

- If `ORACLE_BASE` is not set:

  `ORACLE_HOME``/admin/`*`db_unique_name`*`/`*`pdb_guid`*`/wallet`

2.9.6 Switching Containers

An Enterprise User cannot execute the `ALTER SESSION SET CONTAINER` *`target-CDB`* command, as an enterprise user is not a common user.

3

Configuration and Administration Tools Overview

Configuring Enterprise User Security for an Oracle database primarily involves creating directory objects to store enterprise user and database information. For some implementations, it can also require creating special network configuration files (`ldap.ora`) that enable your databases to locate the correct directory server on the network.

While Oracle Enterprise Manager is your primary tool for both configuring Enterprise User Security and for administration tasks, this chapter introduces all the available tools, in the following topics:

- Enterprise User Security Tools Overview
- Database Configuration Assistant
- Oracle Enterprise Manager
- Oracle Net Configuration Assistant
- Duties of an Enterprise User Security Administrator/DBA

3.1 Enterprise User Security Tools Overview

Enterprise users are database users whose identities are stored and centrally managed in an LDAP directory, such as Oracle Internet Directory. Table 3-1 provides a summary of Enterprise User Security configuration and management tasks and the tools to complete them. The tool names are links to sections that describe them.

Table 3-1 Enterprise User Security Tasks and Tools Summary

Task	Tools
Create users and manage their passwords	Oracle Internet Directory Self-Service Console
Configure databases Oracle home for directory usage over the network	Oracle Net Configuration Assistant
Register and un-register databases in Oracle Internet Directory	Database Configuration Assistant
• Configure enterprise domains and databases in Oracle Internet Directory including mappings, roles and proxy permissions • Manage identity management realm attributes and administrative groups that pertain to Enterprise User Security in Oracle Internet Directory	Oracle Enterprise Manager

Table 3-1 (Cont.) Enterprise User Security Tasks and Tools Summary

Task	Tools
Manage identity management realms in Oracle Internet Directory For information about this tool and realms, refer to *Oracle Identity Management Guide to Delegated Administration*.	Oracle Internet Directory Self-Service Console

3.2 Oracle Internet Directory Self-Service Console

Oracle Internet Directory Self-Service Console is a tool based on Delegated Administration Services. This is a self service application that allows administrated access to the applications data managed in the directory. This tool comes ready to use with Oracle Internet Directory.

The *Oracle Identity Management Guide to Delegated Administration* discusses Delegated Administration Services and the Oracle Internet Directory Self-Service Console tool.

3.3 Oracle Net Configuration Assistant

Oracle Net Configuration Assistant is a wizard-based tool with a graphical user interface. Its primary uses are to configure basic Oracle Net network components, such as listener names and protocol addresses, and to configure your Oracle home for directory server usage. The latter use is what makes this tool important for configuring Enterprise User Security.

If you use Domain Name System (DNS) discovery (automatic domain name lookup) to locate Oracle Internet Directory on your network, then this assistant is not necessary. Note that using DNS discovery is the recommended configuration.

Before you can register a database with the directory, you must do either one of the following two tasks:

- Configure DNS discovery of Oracle Internet Directory on your network.

> **See Also:**
>
> *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about DNS server discovery

- If DNS discovery is not configured on your network, then use Oracle Net Configuration Assistant to create an `ldap.ora` file for your Oracle home.

Your database initially uses the `ldap.ora` file to locate the correct Oracle Internet Directory server on your network. This configuration file contains the hostname, port number, and identity management realm information for your directory server.

Once database registration is complete, the realm is ascertained through the database DN stored in the database wallet.

The following section describes more information about Oracle Net Configuration Assistant: Starting Oracle Net Configuration Assistant.

3.3.1 Starting Oracle Net Configuration Assistant

To start Oracle Net Configuration Assistant:

- (UNIX) From *$ORACLE_HOME*/bin, enter the following at the command line:

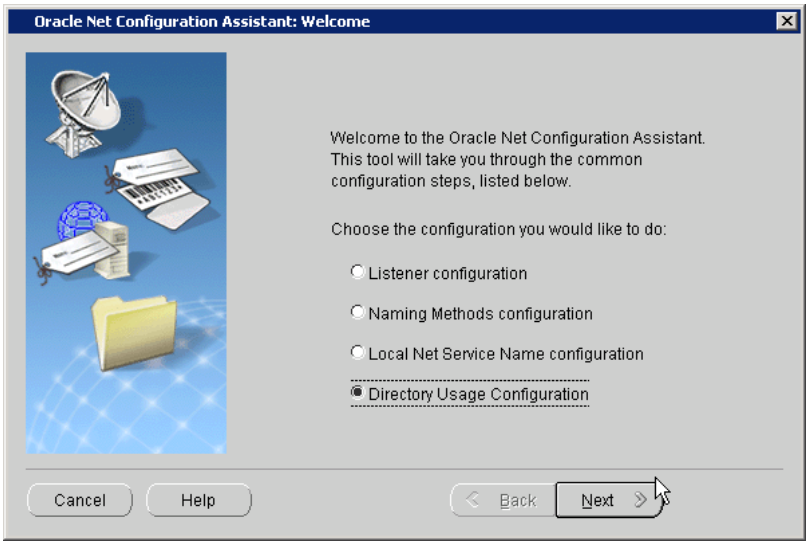
  ```
  netca
  ```

- (Windows) Choose **Start**, **Programs, Oracle-HOME_NAME**, **Configuration and Migration Tools**, **Net Configuration Assistant**

After you start this tool, you will be presented with the opening page shown in Figure 3-1.

Choose the **Directory Usage Configuration** option on this page, click **Next**, and choose the directory server where you wish to store your enterprise users. Then, click **Finish** to create a properly configured `ldap.ora` file for your Oracle home.

Figure 3-1 Opening Page of Oracle Net Configuration Assistant



See Also:

- "Task 5: (Optional) Configure your Oracle home for directory usage" for more information about using this tool to configure your Oracle home for Enterprise User Security
- Oracle Net Configuration Assistant online help and *Oracle Database Net Services Administrator's Guide* for a complete documentation of this tool

3.4 Database Configuration Assistant

Database Configuration Assistant is a wizard-based tool used to create and configure Oracle databases.

Use Database Configuration Assistant to register a database with the directory. In that process, Database Configuration Assistant creates a distinguished name (DN) for the database and the corresponding entry and subtree in Oracle Internet Directory.

The following section describes more information about Database Configuration Assistant: Starting Database Configuration Assistant.

3.4.1 Starting Database Configuration Assistant

To start Database Configuration Assistant:

- (UNIX) From *$ORACLE_HOME*/bin, enter dbca at the command line:
- (Windows) Choose **Start** > **Programs** > **Oracle - HOME_NAME > Configuration and Migration Tools > Database Configuration Assistant**

> **See Also:**
>
> – "To register a database with the directory:" for information about using this tool to register your database
>
> – *Oracle Database Administrator's Guide* for more information about this tool

3.5 The orapki Command-Line Utility

The orapki command-line utility enables administrators to manage wallets, certificate revocation lists, and other public key infrastructure (PKI) elements from the command line. It can be used inside scripts, enabling administrators to automate many routine PKI tasks. The orapki commands enable you to do the following tasks:

Table 3-2 Summary of orapki Commands

Object Affected	Operations Possible with orapki Commands
Certificate	Create or display
CRL (certificate revocation list)	Delete, display, hash, list, or upload
Wallet	Create, display, add, or export

> **See Also:**
>
> *Oracle Database Security Guide* for information about Managing Oracle Database Wallets with the orapki Utility.

3.6 Oracle Enterprise Manager

Enterprise User Security employs Oracle Enterprise Manager to administer enterprise users, administrative groups, enterprise domains, and enterprise roles stored in Oracle Internet Directory. You can use the Web-based user interface provided by Oracle Enterprise Manager to administer Enterprise User Security.

Enterprise users are users provisioned and managed centrally in an LDAP-compliant directory, such as Oracle Internet Directory, for database access. Enterprise domains are directory constructs containing databases, enterprise roles (the access privileges assigned to enterprise users), and proxy permissions (which enable enterprise users to connect to databases as other users).

> **See Also:**
>
> Introducing Enterprise User Security for a discussion of Enterprise User Security administrative groups, enterprise domains, enterprise roles, enterprise users, shared schemas, and user-schema mappings

Use the following steps to access the Enterprise User Security link in Oracle Enterprise Manager Cloud Control:

1. Enter the URL for Cloud Control in a browser window. For example:

   ```
   https://mydbhost:1158/em
   ```

2. Log in as an administrative database user.
3. To navigate to your database, select **Databases** from the **Targets** menu.
4. Click the database name in the list that appears. The database page appears.
5. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
6. Enter the distinguished name (DN) of a directory user, who has administrative privileges for the identity management realm, in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.

3.7 Duties of an Enterprise User Security Administrator/DBA

Enterprise User Security administrators plan, implement, and administer enterprise users. Table 3-3 lists the primary tasks of Enterprise User Security administrators, the tools used to perform the tasks, and the links to where the tasks are documented.

Table 3-3 Common Enterprise User Security Administrator Configuration and Administrative Tasks

Task	Tools Used	See Also
Create an identity management realm in Oracle Internet Directory	Oracle Internet Directory Self-Service Console (Delegated Administration Service)	*Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about how to perform this task
Upgrade an identity management realm in Oracle Internet Directory	Oracle Internet Directory Configuration Assistant	*Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* and the online Help for this tool
Set up DNS to enable automatic discovery of Oracle Internet Directory over the network. Note that this is the recommended configuration.	Oracle Internet Directory Configuration Assistant	*Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* (Domain Name System server discovery) and the online Help for this tool
Create an `ldap.ora` file to enable directory access	Oracle Net Configuration Assistant	"Task 5: (Optional) Configure your Oracle home for directory usage"
Register a database in the directory	Database Configuration Assistant	"Task 6: Register the database in the directory"
Configure password authentication for Enterprise User Security	Oracle Enterprise Manager	"Configuring Enterprise User Security for Password Authentication"
Configure Kerberos authentication for Enterprise User Security	• Oracle Internet Directory Self-Service Console (Delegated Administration Service) • Oracle Enterprise Manager	"Configuring Enterprise User Security for Kerberos Authentication"
Configure SSL authentication for Enterprise User Security	• Oracle Net Manager • Oracle Enterprise Manager	"Configuring Enterprise User Security for SSL Authentication"
Create or modify user entries and Oracle administrative groups in the directory	Oracle Internet Directory Self-Service Console (Delegated Administration Service)	• "Administering Identity Management Realms" • "Administering Enterprise Users"
Create or modify enterprise roles and domains in the directory	Oracle Enterprise Manager	• "Administering Enterprise Domains" • "Configuring Enterprise Roles"
Create or modify wallets for directory, databases, and clients	orapki command line utility	• *Oracle Database Security Guide* for information about `orapki` Utility
Change a user's database or directory password	Oracle Internet Directory Self-Service Console (Delegated Administration Service)	"Setting Enterprise User Passwords"
Change a database's directory password	Database Configuration Assistant	"To change the database's directory password:"
Request initial Kerberos ticket when KDC is not part of the operating system, such as Kerberos V5 from MIT	`okinit` utility	*Oracle Database Security Guide* for information about using the `okinit` utility to get an initial Kerberos ticket

4

Enterprise User Security Configuration Tasks and Troubleshooting

This chapter describes configuring Enterprise User Security using a sequence of steps. They include the initial database and directory preparation through connecting to the database as an enterprise user, where authentication can use passwords, Kerberos tickets, or SSL. A troubleshooting section helps you when you test your Enterprise User Security implementation.

This chapter contains the following topics:

4.1 Enterprise User Security Configuration Overview

Configuring Enterprise User Security means creating shared schemas and global roles in databases that you want accessible to enterprise users. You configure the identity management realm in the directory to reflect those database roles and schemas, and then associate directory users with them. These steps apply regardless of the authentication method you choose: password, Kerberos, or SSL.

The primary configuration differences among the authentication types are in network connection configuration. You must consider the following three connection types:

- Client-to-database
- Database-to-directory

Enterprise User Security supports many combinations of authentication types between databases, directories, and clients. The three most common implementations of Enterprise User Security, described in this chapter, use the following authentication methods for client-database and database-directory connections:

- Passwords for both connections
- SSL for both connections
- Kerberos for client-database connections and passwords for database-directory connections

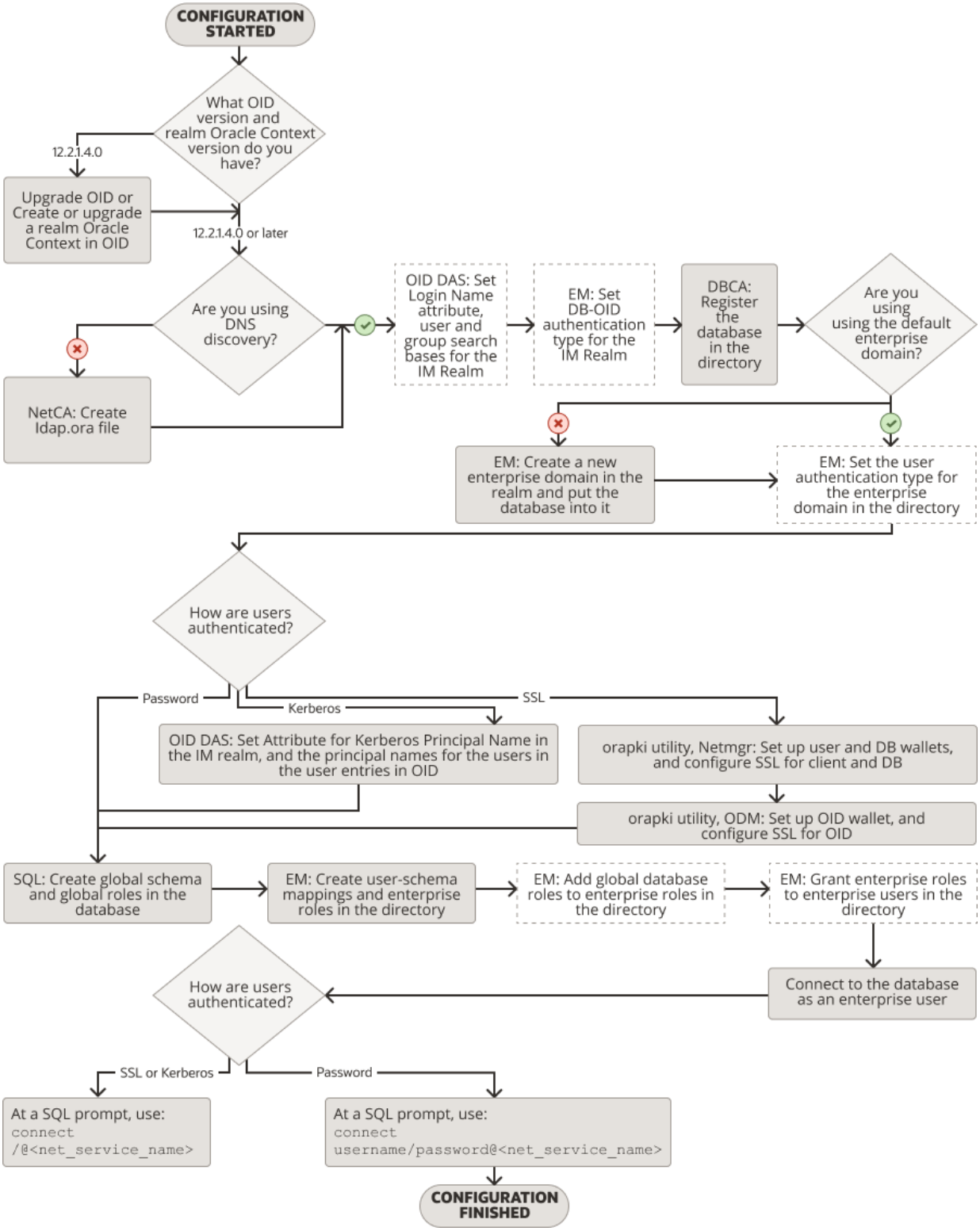
You decide which of these to choose based primarily on your network environment, because the security and integrity of your enterprise data depend on creating secure network connections. Typical network environments can have all clients, databases, and directories

residing within the same network behind a firewall, or distributed across several networks and perhaps exposed to the Internet. Different environments can dictate what authentication types you choose, in order to secure your Enterprise User Security network connections.

A second consideration in making such choices is the fact that more rigorous authentication types, such as SSL and Kerberos, require greater configuration complexity, additional software, and ongoing maintenance.

Figure 4-1 shows the configuration process for Enterprise User Security. It is a step-by-step process with decision points based on your implementation and how your users are authenticated. The configuration steps represented with broken lines are optional.

Figure 4-1 Enterprise User Security Configuration Flow Chart



For brevity, some product names and features have been abbreviated in this flow chart. The following table lists the abbreviations used and the meaning of each:

Abbreviation	Meaning
DBCA	Database Configuration Assistant
EM	Oracle Enterprise Manager
IM Realm	Identity Management Realm
Netmgr	Oracle Net Manager
ODM	Oracle Directory Manager
OID	Oracle Internet Directory
OID DAS	Oracle Internet Directory Delegated Administration Services
SQL	SQL*Plus

See Also:

Introducing Enterprise User Security for information about the realm Oracle Context, its administrative groups, and entries that pertain to Enterprise User Security

4.2 Enterprise User Security Configuration Roadmap

This section provides detailed descriptions of the configuration steps that Figure 4-1 illustrates. They should be performed in the following order:

1. "Preparing the Directory for Enterprise User Security (Phase One)"
2. "Configuring Enterprise User Security Objects in the Database and the Directory (Phase Two)"
3. "Configure Enterprise User Security for the Authentication Method You Require (Phase Three)", which completes your Enterprise User Security configuration by establishing your chosen authentication method as one of the following three:
   - "Configuring Enterprise User Security for Password Authentication"
   - "Configuring Enterprise User Security for Kerberos Authentication"
   - "Configuring Enterprise User Security for SSL Authentication"

4.3 Preparing the Directory for Enterprise User Security (Phase One)

This configuration phase must be performed before you can configure any other part of Enterprise User Security.

Enterprise User Security for Release 12*c* or later requires Release 9.0.4 (or later) version of Oracle Internet Directory, which installs with the required version of the Oracle schema. This schema is backward compatible. After you have installed Oracle Internet Directory, perform the following directory usage configuration tasks:

- Task 1: (Optional) Create an identity management realm in the directory

- Task 2: (Optional) Set identity management realm properties
- Task 3: Identify administrative users in the directory
- Task 4: (Optional) Set the default database-to-directory authentication type for the identity management realm
- Task 5: (Optional) Configure your Oracle home for directory usage
- Task 6: Register the database in the directory

Task 1: (Optional) Create an identity management realm in the directory

If necessary, use Oracle Internet Directory Self-Service Console (Delegated Administration Service) to create an identity management realm in the directory. You can use Oracle Internet Directory Configuration Assistant to upgrade an Oracle9*i* Oracle Context to a 9.0.4 or higher version Identity Management Realm.

You must have version 9.0.4 (or later) identity management realm to use Oracle Database 10*g* or Oracle Database 11*g*. Version 9.0.4 realms are backward compatible to Oracle9*i*, so you can register Oracle9*i* and Oracle Database 12*c* Release 1 (12.1) in the same realm and place them in the same domain, if desired.

> **See Also:**
>
> *Oracle Identity Management Guide to Delegated Administration* for more information on creating identity management realms in Oracle Internet Directory

Task 2: (Optional) Set identity management realm properties

Table 4-1 shows the defaults for a version 9.0.4 identity management realm.

Table 4-1 Identity Realm Defaults

User Search Base	Group Search Base	Login Name Attribute (nickname)
`cn=Users,`*`realm_DN`*	`cn=Groups,`*`realm_DN`*	*`uid`*, the user id

If you want different settings, then use Oracle Internet Directory Self-Service Console to set the user search base, group search base, and login name attribute (nickname). You can also set up the necessary context administrators in the identity management realm you plan to use in the directory.

To perform this task, see "Setting Properties of an Identity Management Realm".

> **Note:**
>
> Each identity management realm includes an orcladmin user who is the root user of that realm only. These realm-specific orcladmin users are represented by the directory entries `cn=orcladmin,cn=Users,<realm_DN>`. Note that when you are logged in to Enterprise User Security administration tools as a realm-specific orcladmin user, then you can only manage directory objects for that realm. To manage objects in another realm, you must log in to administration tools as the orcladmin user for that realm.

Task 3: Identify administrative users in the directory

Identify administrative users in the directory who are authorized to perform the following tasks:

- Register databases
- Administer database security
- Create and manage enterprise domains

If administrative users do not already exist who can perform these tasks, then see Administering Enterprise User Security to create them.

> **Note:**
>
> Although one administrator can perform all Enterprise User Security administrative tasks, you can create many different kinds of administrators so security tasks can be assigned to different people. Separating security tasks in this way results in a more secure enterprise environment, but this requires coordination among the different administrators.

Task 4: (Optional) Set the default database-to-directory authentication type for the identity management realm

By default, the database-to-directory authentication type for the identity management realm is set to passwords. If you want a different default setting, then use the Oracle Enterprise Manager interface to change it. For example, if you are using a public key infrastructure (PKI), then you would need to set the authentication type to SSL. See "Setting the Default Database-to-Directory Authentication Type for an Identity Management Realm".

> **Note:**
>
> - This default realmwide setting can be overridden on a database by setting the `LDAP_DIRECTORY_ACCESS` initialization parameter. See *Oracle Database Reference* for more information about this parameter.
> - If you are using SSL, then see *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about setting up SSL with two-way authentication for Oracle Internet Directory.

Task 5: (Optional) Configure your Oracle home for directory usage

This step is optional because users of Domain Name System (DNS) discovery (automatic domain name lookup to locate the directory on a network) do not need to perform this step. (See *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about DNS server discovery.)

If you are *not* using DNS discovery, then you *must* use Oracle Net Configuration Assistant (NetCA) to create an `ldap.ora` file for your Oracle home. This configuration file specifies the directory host and port information, and the location of the identity management realm so the database can connect to the directory. (See "Starting Oracle Net Configuration Assistant")

To create an `ldap.ora` file for your Oracle home:

1. In the Oracle Net Configuration Assistant welcome page, choose **Directory Service Usage Configuration**, and click **Next**.
2. On the Directory Usage Configuration page, select an option appropriate for your environment. Then follow the prompts in the wizard and refer to the online Help to create an `ldap.ora` file for your Oracle home.

> **Note:**
>
> - SSL authentication between your database and directory requires that the SSL port entered in the `ldap.ora` file support two-way authentication, in which both client and server send certificates to each other. Thus, you must acquire a PKI digital certificate and wallet for Oracle Internet Directory, and bring up Oracle Internet Directory in the SSL mutual authentication mode. The second port in the `ldap.ora` file should have the SSL mutual authentication port. (See *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.)
> - If you are using password authentication for your database-to-directory connection, then the SSL port entered in the `ldap.ora` file must support SSL with no authentication. No wallet or certificate is required for Oracle Internet Directory. The second port in the `ldap.ora` file should have the SSL no authentication port.

> **See Also:**
>
> "Configuring Your Database to Use the Directory" for an example of using NetCA to configure directory usage

Task 6: Register the database in the directory

After you have configured your Oracle home for directory usage, use Database Configuration Assistant to register the database in the directory. Registration creates an entry in the directory so the database can bind (log in) to it.

> **Note:**
>
> To perform this task, you must be the directory superuser or a member of either the `OracleDBCreators` group or the `OracleContextAdmins` group.

When registering a database in the directory, Database Configuration Assistant performs the following configuration tasks:

- Creates a new database service entry and subtree, and assigns a DN to it in the Oracle Context for the identity management realm you are using.
- Adds the database to the default enterprise domain.
- Establishes the authentication type of the database to the directory by setting the `LDAP_DIRECTORY_ACCESS` parameter to one of the three allowable settings: `NONE`, `PASSWORD`, or `SSL`. Database Configuration Assistant reads the default database to directory authentication attribute setting for the identity management realm to determine the authentication type setting for the database.

  The `LDAP_DIRECTORY_ACCESS` parameter, residing in the database initialization parameter file, determines whether and how the database attempts authentication to the directory. An administrator can change this authentication type setting by using the `ALTER SYSTEM` command.
- Creates a database wallet, containing the database DN in the following form:

  `cn=`*`short_database_name`*`,cn=OracleContext,`*`realm_DN`*

  where *`short_database_name`* is the first part of the fully qualified domain name for a database.

  For example, if you have a database named `db1.us.example.com`, then the short database name is `db1`.
- Randomly generates a database password for directory access, storing it in the database wallet and in the directory.
- After creating the wallet, Database Configuration Assistant stores it at `$ORACLE_BASE/admin/`*`Oracle_SID`*`/wallet` (in UNIX environments), if the `ORACLE_BASE` environment variable is present. If the `ORACLE_BASE` environment variable is not present, then the `$ORACLE_HOME/admin/`*`Oracle_SID`*`/wallet` directory is used.

  In Windows environments, replace the slashes (/) with backslashes (\).

If a database wallet already exists, then Database Configuration Assistant uses it and updates the password in the wallet.

- Enables autologin for the database wallet.

> **Note:**
>
> The database's password-based credentials for authentication to Oracle Internet Directory are placed in the wallet when an Oracle database is registered in Oracle Internet Directory.

To register a database with the directory:

See "Starting Database Configuration Assistant" to start this tool.

1. After starting Database Configuration Assistant, select **Configure Database Options in a Database** and click **Next**.
2. Select a database and click **Next**.
3. To register the database, click **Yes, Register the Database**.
4. Enter a Custom Database Name for the database.

   The ability to specify a custom database name was introduced in Oracle Database 12*c* Release 1 (12.1). By default, the database CN (first part of the DN or the distinguished name) in the directory is the *DB_UNIQUE_NAME*. You can change this to a custom value.
5. Enter the directory credentials for a user in the OracleDBCreators group.
6. Enter a password for the database wallet.

   > **Note:**
   >
   > Remember the database wallet password you entered in Step 5. It cannot be retrieved after you finish database registration. If you do not know the password, a multistep process is required to generate a new wallet and reregister the database. See "About the Database Wallet and Password" for further information.
7. Click **Finish** if you are only registering the database. Click **Next** if you want to configure additional database features.

   > **See Also:**
   >
   > "Registering Your Database with the Directory" for an example of using DBCA to register the database

To change the database's directory password:

After starting Database Configuration Assistant, select **Configure Database Options in a Database**, and click **Next**.

1. Select a database and click **Next**.
2. Select **Regenerate database password**.
3. Enter the directory credentials for a user in the OracleDBCreators group and a password for the database wallet. Click **OK**.
4. Click **Finish** if you are only regenerating the password. Click **Next** if you want to configure additional database features.

To unregister a database from the directory:

See "Starting Database Configuration Assistant" to start this tool.

1. After starting Database Configuration Assistant, select **Configure Database Options in a Database** and click **Next**.
2. Select a database and click **Next**.
3. To unregister the database, select the **Unregister** option.
4. Enter the directory credentials for a user with the appropriate permissions.
5. Enter a password for the database wallet.

When you unregister a database from the directory, Database Configuration Assistant performs the following configuration tasks:

- Removes the database entry and subtree from the directory.
- Sets the `LDAP_DIRECTORY_ACCESS` parameter to `NONE`.
- Removes the database from its enterprise domain (if the user has sufficient permissions).

> **Note:**
>
> Depending on user permissions, Database Configuration Assistant may be unable to remove a database from its domain in the directory. If it cannot, then use Oracle Enterprise Manager to remove it from the enterprise domain.

- Does not remove the database wallet.

> **Note:**
>
> To succeed at unregistering an Oracle Database from Oracle Internet Directory by using Database Configuration Assistant, you must be one of the following:
>
> - A member of the Oracle Context Admin group
> - A member of both the Database Admin group (for the database you are unregistering) and the Database Security Admin group
> - A member of both the Database Admin group (for the database you are unregistering) and the Domain Admin group (for the enterprise domain that contains the database).

This section includes the following topics:

- About the Database Wallet and Password
- Configuring Directory Access for Enterprise Users

4.3.1 Configuring Directory Access for Enterprise Users

You can configure the Oracle Internet Directory services connection manually by using LDAP-specific Oracle Database system parameters.

1. Ensure that you have created the `dsi.ora` file or the `ldap.ora` file, and that you have created the wallet.
2. Log in to the appropriate PDB as a user who has the `ALTER SYSTEM` system privilege.

   For example:

   ```
   sqlplus sec_admin@pdb_name
   Enter password: password
   ```

   To find the available PDBs in a CDB, log in to the CDB root container and then query the `PDB_NAME` column of the `DBA_PDBS` data dictionary view. To check the current container, run the `show con_name` command.
3. Modify the `LDAP_DIRECTORY_ACCESS` parameter, which determines the type of LDAP directory access.

   Set `LDAP_DIRECTORY_ACCESS` in each PDB, not in the CDB root. Setting this parameter in the CDB root will apply it only to the root, not to the PDBs.

   Valid values are `PASSWORD`, `SSL` and `NONE` (to disable the connection). If you set this parameter to `NONE`, enterprise users from Oracle Internet Directory cannot log in to Oracle database.

   For example:

   ```
   ALTER SYSTEM SET LDAP_DIRECTORY_ACCESS = 'PASSWORD';
   ```

   You can also set this parameter in the spfile or in the `init.ora` file (if the `init.ora` file is used). Afterward, restart the database.
4. Set the `LDAP_DIRECTORY_SYSAUTH` parameter to `YES`, so that administrative users can log in to Oracle Database with the `SYSDBA`, `SYSOPER`, `SYSBACKUP`, `SYSDG`, `SYSKM`, or `SYSRAC` administrative privilege.

   Set `LDAP_DIRECTORY_SYSAUTH` in each PDB, not in the CDB root. Setting this parameter in the CDB root will apply it only to the root, not to the PDBs.

   If you set this parameter to `NO`, then enterprise users from Oracle Internet Directory cannot log in to Oracle database with these privileges.

   ```
   ALTER SYSTEM SET LDAP_DIRECTORY_SYSAUTH = YES SCOPE=SPFILE ;
   ```

   You can also set this parameter in the spfile or in the `init.ora` file (if the `init.ora` file is used). Afterward, restart the database.
5. Connect to the root as a user with the `SYSDBA` administrative privilege.
6. Close and then re-open the PDB.

   ```
   ALTER PLUGGABLE DATABASE pdb_name CLOSE IMMEDIATE;
   ALTER PLUGGABLE DATABASE pdb_name OPEN;
   ```

After you re-open the PDB, you can log in to the PDB with the `SYSDBA` administrative privilege and check the LDAP parameters settings as follows:

```
show parameter ldap
```

Related Topics

- LDAP_DIRECTORY_SYSAUTH
- LDAP_DIRECTORY_ACCESS

4.3.2 About the Database Wallet and Password

The database requires the wallet even if no SSL (Secure Sockets Layer) is used to secure the connection between the database and the directory. If SSL is used, then this wallet should be used to store the database's digital PKI certificate.

The wallet password you enter when using Database Configuration Assistant to register a database in the directory is the password to the wallet itself. This password is not the database's directory login credentials.

You can change this wallet password later, using the `orapki` utility. If you forget this wallet password, then you must generate an entirely new wallet and password. To do so, you must first delete the existing database wallet, create a new wallet (which can be empty) and put it at the default wallet location, `$ORACLE_HOME/admin/`*`Oracle_SID`*`/wallet` (in UNIX environment). Next, unregister the database from the directory, and reregister the database in the directory. During that registration, another database wallet and password can be generated.

After you have prepared the directory for Enterprise User Security, then you can create the Enterprise User Security database and directory objects as described in "Configuring Enterprise User Security Objects in the Database and the Directory (Phase Two)".

> **See Also:**
>
> - *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about configuring an identity management realm in the directory
> - *Oracle Database Reference* for information about changing the value of the `LDAP_DIRECTORY_ACCESS` initialization parameter

This section includes the following topic: Sharing Wallets and sqlnet.ora Files Among Multiple Databases.

4.3.2.1 Sharing Wallets and sqlnet.ora Files Among Multiple Databases

Multiple databases (that are not replicas) cannot share wallets, because wallets contain a database's identity. Therefore, if a `sqlnet.ora` file contains a wallet location, then multiple databases cannot share that `sqlnet.ora` file.

In order to share a single `sqlnet.ora` file among multiple databases, the following preconditions are required:

- User authentication should use passwords or Kerberos.
- The wallet containing the password should reside at the default wallet location, which is where Database Configuration Assistant creates it.

If the preceding conditions are met, then multiple databases *can* share the `sqlnet.ora` file because no wallet location information is stored in that file.

However, when *SSL* authentication is used between the user (client) and the database, the wallet location must be specified in the database server's `sqlnet.ora` file. Such a `sqlnet.ora` file cannot be shared by multiple databases for SSL-authenticated enterprise users.

4.4 Configuring Enterprise User Security Objects in the Database and the Directory (Phase Two)

This is the second phase of configuration steps required to implement Enterprise User Security. The configuration steps in this section assume the following recommended setup:

- You have prepared your database and your directory by completing the tasks described in "Preparing the Directory for Enterprise User Security (Phase One)".
- Your users are stored in an identity management realm Users subtree.
- You use the OracleDefaultDomain, which is the default enterprise domain that Database Configuration Assistant uses when you register databases in the directory.

Note that databases must be in an enterprise domain that is in an identity management realm in order for enterprise user logins to work.

> **See Also:**
>
> If you do not use the OracleDefaultDomain or store your users in an identity management realm Users subtree, then see the following documentation:
>
> - *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about creating a new identity management realm or modifying an existing one, and for information about setting access control lists on directory objects
> - "Creating an Enterprise Domain" to create another domain in which to put your database. Then substitute your new domain name for OracleDefaultDomain in the following configuration steps

To configure Enterprise User Security objects in the database and directory perform the following tasks:

- Task 1: Create Global Schemas and Global Roles in the Database
- Task 2: Configure User-Schema Mappings for the Enterprise Domain
- Task 3: Create Enterprise Roles in the Enterprise Domain
- Task 4: Add Global Database Roles to Enterprise Roles
- Task 5: Grant Enterprise Roles to Enterprise Users for Database Access

Task 1: Create Global Schemas and Global Roles in the Database

Although this step can also be completed by using Oracle Enterprise Manager, the following examples use SQL*Plus directly:

1. Create a shared schema for enterprise users. The following syntax example creates a shared schema named guest:

   ```
   SQL> CREATE USER guest IDENTIFIED GLOBALLY AS '';
   ```

   If you do not want to use a shared schema, then specify a user DN between the single quotation marks to create an exclusive schema.

2. Grant the `CREATE SESSION` privilege to the shared schema created in Step 1 so users can connect to it. The following syntax example grants the `CREATE SESSION` privilege to the guest shared schema:

   ```
   SQL> GRANT CREATE SESSION TO guest;
   ```

   Alternatively, you can grant the `CREATE SESSION` privilege to a global role, which you grant to specific users through an enterprise role. See Step 3.

3. Create global roles for the database to hold relevant privileges. The following syntax examples create the `emprole` and `custrole` global roles:

   ```
   SQL> CREATE ROLE emprole IDENTIFIED GLOBALLY;
   SQL> CREATE ROLE custrole IDENTIFIED GLOBALLY;
   ```

   Global roles are associated with enterprise roles, which are created later, and then are allocated to enterprise users.

4. Grant privileges to the new global roles that were created in Step 3. The following syntax example grants the `SELECT` privilege to `emprole` and `custrole` global roles on the products table:

   ```
   SQL> GRANT select ON products TO custrole, emprole;
   ```

> **See Also:**
>
> *Oracle Database SQL Language Reference* for information about the syntax for creating a user.
>
> *Oracle Database SQL Language Reference* for information about the syntax for granting a privilege to a user or role.
>
> *Oracle Database SQL Language Reference* for information about the syntax for creating a role.

Task 2: Configure User-Schema Mappings for the Enterprise Domain

Use Enterprise Manager to configure user-schema mappings for the OracleDefaultDomain by using the following steps:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.

4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Manage Enterprise Domains**.

   The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.
7. Select **OracleDefaultDomain**. Click **Configure**.

   The Configure Domain page appears.
8. Click the **User-Schema Mappings** tab. All user-schema maps created at the domain level are displayed. User-schema maps created at database levels are not displayed here.
9. Click **Create** to create a new user-schema mapping for the domain.

   The Create Mapping page is displayed.
10. Under the From section, select **Users** to map an individual enterprise user to a database schema. Alternatively, select **Subtree** to map a directory subtree containing multiple users. You can use the Search icon to search for the appropriate user or subtree.
11. Under the To section, enter the name of the **Schema** to which the user or subtree should be mapped. This is the schema that you created in Task 1.
12. Click **Continue** in the Create Mapping page.
13. Click **OK** in the Configure Domain page.

> **Note:**
>
> You can also create user-schema mappings for an individual database in an enterprise domain. Such mappings apply only to that particular database and not to other databases in the domain.

> **See Also:**
>
> "Mapping Enterprise Users to the Shared Schema" for an example on creating user-schema mappings

Task 3: Create Enterprise Roles in the Enterprise Domain

Use Enterprise Manager to create enterprise roles in the OracleDefaultDomain by using the following steps:

1. Select OracleDefaultDomain in the Manage Enterprise Domains page. Click **Configure**.

   The Configure Domain page appears.
2. Click the **Enterprise Roles** tab.
3. Click **Create** to create a new enterprise role.

The **Create Enterprise Role** page appears.

4. Enter a name for the enterprise role in the **Name** field. Click **Continue**.

   The new role is displayed in the Configure Domain page.

> **See Also:**
>
> "Using Enterprise Roles" for an example on creating and using enterprise roles

Task 4: Add Global Database Roles to Enterprise Roles

Use Enterprise Manager to add the global database roles that you created in Task 1 to the enterprise roles that you created in Task 3 by using the following steps:

1. Select the enterprise role that you just created in the Configure Domain page. Click **Edit**.

   The Edit Enterprise Role page is displayed.

2. Make sure that the **DB Global Roles** tab is selected. Click **Add** to add global roles from databases that are part of the enterprise domain.

   The Search and Select Database Global Roles page appears.

3. Select the **Database** that contains the global roles you wish to add. Log in to the selected database by supplying a **User Name** and **Password**. Click **Go**.

4. Select the global roles to add. Click **Select**.

   The selected roles appear in the Edit Enterprise Role page.

> **See Also:**
>
> "Using Enterprise Roles" for an example on creating and using enterprise roles

Task 5: Grant Enterprise Roles to Enterprise Users for Database Access

Use Enterprise Manager to grant enterprise roles that you created in Task 3 to the enterprise users by using the following steps:

1. Click the **Grantees** tab in the Edit Enterprise Role page.

2. Click **Add**.

   The Select Users or Groups page is displayed.

3. Select the **Search Base** or the subtree that contains the user or group. Select User under **View** if you are granting the enterprise role to a user. Select Group under **View**, if you are granting the role to a group. Optionally, enter the common name of the user or group in the **Name** field. Click **Go**.

4. Select the users or groups to be granted the enterprise role. Click **Select**.

5. Click **Continue** in the Edit Enterprise Role page.

6. Click **OK** in the Configure Domain page.

> **See Also:**
>
> "Using Enterprise Roles" for an example on creating and using enterprise roles

4.5 Configure Enterprise User Security for the Authentication Method You Require (Phase Three)

In the third phase, you complete the Enterprise User Security configuration based on the authentication method you have chosen. Go to one of the following sections:

- "Configuring Enterprise User Security for Password Authentication"
- "Configuring Enterprise User Security for Kerberos Authentication"
- "Configuring Enterprise User Security for SSL Authentication"

> **See Also:**
>
> Table 1-1 for a comparison of the benefits provided by password, Kerberos, and SSL authentication for Enterprise User Security

4.5.1 Configuring Enterprise User Security for Password Authentication

By default, new enterprise domains are configured to accept all supported user authentication types (password, Kerberos, and SSL). If you want enterprise users to be authenticated by passwords, then you must configure that as described in the following tasks.

The configuration steps in this section assume the following:

- You have prepared your directory by completing the tasks described in "Preparing the Directory for Enterprise User Security (Phase One)".
- You have configured your Enterprise User Security objects in the database and the directory by completing the tasks described in "Configuring Enterprise User Security Objects in the Database and the Directory (Phase Two)".
- You have configured an SSL instance with no authentication for Oracle Internet Directory as described in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*. If you are using an `ldap.ora` file, then also ensure that the port number for this SSL with no authentication instance is listed there as your directory SSL port.

To configure Enterprise User Security for password authentication, perform the following tasks:

- Task 1: (Optional) Enable the Enterprise Domain to Accept Password Authentication
- Task 2: Connect as a Password-Authenticated Enterprise User

Task 1: (Optional) Enable the Enterprise Domain to Accept Password Authentication

By default, OracleDefaultDomain is configured to accept password authentication. If this has been changed, then use Oracle Enterprise Manager to enable password authentication for OracleDefaultDomain using the following steps:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Manage Enterprise Domains**.

   The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.
7. Select **OracleDefaultDomain**. Click **Configure**.

   The Configure Domain page appears.
8. Click the **Configuration** tab.
9. Under User Authentication Types Accepted, select **Password**.
10. Click **OK**.

Task 2: Connect as a Password-Authenticated Enterprise User

For an enterprise user whose directory login name is `hscortea` and whose password is `Easy2rem`, enter the following to connect to the database by using SQL*Plus:

```
SQL> connect hscortea@<Oracle Net Service Name>
Enter password:
/* Enter Easy2rem when prompted for the password*/
```

The database authenticates the enterprise user (`hscortea`) by verifying the username-password combination against the directory entry associated with this user. Then, it identifies the proper schema and retrieves the user's global roles. If successful, then the connection to the database is established.

If your connection succeeds, then the system responds `Connected to:...`. This is the confirmation message of a successful connect and setup. If an error message is displayed, then see "ORA-n Errors for Password-Authenticated Enterprise Users".

If you do connect successfully, then check that the appropriate global roles were retrieved from the directory, by entering the following at the SQL*Plus prompt:

```
select * from session_roles
```

If the global roles were not retrieved from the directory, then see "NO-GLOBAL-ROLES Checklist".

You have completed password-authenticated Enterprise User Security configuration.

> **See Also:**
>
> - "Troubleshooting Enterprise User Security" for information about diagnosing and resolving errors
> - Administering Enterprise User Security for information about configuring the identity management realm, and about creating and managing enterprise domains, enterprise roles, and enterprise users

4.5.2 Configuring Enterprise User Security for Kerberos Authentication

The configuration steps in this section assume the following:

- You have registered your databases with the Kerberos authentication server and configured your Oracle Net Services as described in information about configuring Kerberos authentication in *Oracle Database Security Guide*.
- You have prepared your directory by completing the tasks described in "Preparing the Directory for Enterprise User Security (Phase One)".
- You have configured your Enterprise User Security objects in the database and the directory by completing the tasks described in "Configuring Enterprise User Security Objects in the Database and the Directory (Phase Two)".
- You have configured an SSL instance with no authentication for Oracle Internet Directory as described in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*. If you are using an `ldap.ora`, then also ensure that the port number for this SSL with no authentication instance is listed there as your directory SSL port.

To configure Enterprise User Security for Kerberos authentication, perform the following tasks:

- Task 1: Configure Oracle Internet Directory Self-Service Console to display the Kerberos principal name attribute
- Task 2: (Optional) Configure the Kerberos Principal Name Directory Attribute for the Identity Management Realm
- Task 3: Specify the Enterprise User's Kerberos Principal Name in the krbPrincipalName Attribute
- Task 4: (Optional) Enable the Enterprise Domain to Accept Kerberos Authentication
- Task 5: Connect as a Kerberos-Authenticated Enterprise User

Task 1: Configure Oracle Internet Directory Self-Service Console to display the Kerberos principal name attribute

By default, the Oracle Internet Directory Self-Service Console user interface does not display the field where you can configure Kerberos principal names. The first time you create Kerberos-authenticated users in the directory, you must configure this tool to display the `krbPrincipalName` attribute in its Create User page by using the following steps:

1. Log in to the Oracle Internet Directory Self-Service Console.

Enter the URL to access the Oracle Internet Directory Self-Service Console in a browser window. For example:

```
http://myhost1:7777/oiddas
```

Log in as the `orcladmin` user.

2. Click the **Configuration** tab. Click the **User Entry** subtab.
3. Click **Next** until the Configure User Attributes page appears.
4. In the Configure User Attributes page, click **Add New Attribute**.

   The Add New Attribute page appears.
5. In the Add New Attribute page, select **krbPrincipalName** from the **Directory Attribute Name** box (or the attribute that you have configured for `orclCommonKrbPrincipalAttribute` in your identity management realm) and perform the following steps on this page:
   a. Enter a value, say Kerberos Principal Name, for the **UI Label**.
   b. Select **Searchable** and **Viewable**.
   c. Select **Single Line Text** from the **UI Type**.
   d. Click**Done**.
6. Click **Next** to navigate to the Configure Attribute Categories page. Select **Basic Information** and click **Edit**.

   The Edit Category page appears.
7. Perform the following steps on the **Edit Category** page:
   a. Select **krbPrincipalName** in the left category list.
   b. Click **Move,** to move **krbPrincipalName** to the right-hand list.
   c. Click **Done**.
8. Click **Next** until you reach the last step. Click **Finish** to save your work.

Task 2: (Optional) Configure the Kerberos Principal Name Directory Attribute for the Identity Management Realm

Use Oracle Internet Directory Self-Service Console to enter the directory attribute used to store the Kerberos principal name for the identity management realm you are using in the directory. By default, Kerberos principal names are stored in the `krbPrincipalName` attribute but can be changed to correspond to your directory configuration by changing `orclCommonKrbPrincipalAttribute` in the identity management realm. For more information about this task, see "Setting Login Name, Kerberos Principal Name, User Search Base, and Group Search Base Identity Management Realm Attributes".

> **Note:**
>
> By default, the Oracle Internet Directory Self-Service Console user interface does not display the field where you can configure Kerberos principal names. The first time you create Kerberos-authenticated users in the directory, you must configure the console to display the `krbPrincipalName` attribute in its Create User window.

Task 3: Specify the Enterprise User's Kerberos Principal Name in the krbPrincipalName Attribute

Use Oracle Internet Directory Self-Service Console to specify the enterprise user's Kerberos principal name (`Kerberos_username@Kerberos_realm`) in the `krbPrincipalName` attribute of the enterprise user's directory entry. For more information about this task, see "Creating New Enterprise Users".

Task 4: (Optional) Enable the Enterprise Domain to Accept Kerberos Authentication

By default, OracleDefaultDomain is configured to accept all types of authentication. If this has been changed or if you are using another domain, then use Oracle Enterprise Manager to enable Kerberos authentication for your enterprise domain by performing the following steps:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Manage Enterprise Domains**.

   The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.
7. Select **OracleDefaultDomain**. Click **Configure**.

   The Configure Domain page appears.
8. Click the **Configuration** tab.
9. Under User Authentication Types Accepted, select **Kerberos**.
10. Click **OK**.

Task 5: Connect as a Kerberos-Authenticated Enterprise User

If the KDC is not part of the operating system, such as Kerberos V5 from MIT, then the user must get an initial ticket with the `FORWARDABLE` flag set by using the `okinit` utility. See *Oracle Database Security Guide* for information about obtaining the initial ticket with the `okinit` Utility.

If the KDC is part of the operating system, such as Windows 2000 or some versions of Linux or UNIX, then the operating system automatically picks up the user's ticket (with the `FORWARDABLE` flag set) from the cache when the user logs in.

The user connects to the database by launching SQL*Plus and entering the following at the command line:

```
SQL> connect /@<net_service_name>
```

The database uses Kerberos to authenticate the user. The database authenticates itself to the directory by password.

If your connection succeeds, then the system responds with `Connected to:...`. This is the confirmation message of a successful connect and setup. If an error message is displayed, then see "ORA-n Errors for Kerberos-Authenticated Enterprise Users".

If you do connect successfully, then check that the appropriate global roles were retrieved from the directory, by entering the following at the SQL*Plus prompt:

```
select * from session_roles
```

If the global roles were not retrieved from the directory, then see "NO-GLOBAL-ROLES Checklist".

You have completed Kerberos-authenticated Enterprise User Security configuration.

See Also:

- "Troubleshooting Enterprise User Security" for information about diagnosing and resolving errors
- Administering Enterprise User Security for information about configuring the identity management realm, and information about creating and managing enterprise domains, enterprise roles, and enterprise users

4.5.3 Configuring Enterprise User Security for SSL Authentication

The configuration steps in this section assume the following:

- You have obtained the appropriate PKI credentials and used `orapki` utility to create wallets for the directories, databases, and clients that you want to include in your Enterprise User Security implementation.
- You have confirmed that each enterprise user entry in Oracle Internet Directory is provisioned with a unique PKI credential. However, in this release an enterprise user can have different DNs in their PKI certificate and Oracle Internet Directory entry. Also in this release, the database entry can have different DNs in its PKI certificate and Oracle Internet Directory entry.

  You must provision user certificates in their respective Oracle Internet Directory user entries in order to support using different DNs in the certificate and the directory. A user certificate is provisioned in to the `usercertificate` attribute of the user entry. If you prefer not to provision the certificates, then you must make sure that the subject DNs in the certificates match the user DNs in the directory.

  Oracle Internet Directory 10*g* Release2 (10.1.2) includes certificate matching rules to support the new functionality of being able to use different DNs in the certificate and the directory. The `orclpkimatchingrule` attribute in Oracle Internet Directory determines the type of match that is used.

The default value of `orclpkimatchingrule` is 2. This enables you to support both provisioned and non-provisioned user entries. The database finds out a user's Oracle Internet Directory DN based on a search for the user's certificate provisioned in the directory. If the certificate search fails, then the database reverts to using an exact match between the user's certificate DN and their Oracle Internet Directory DN.

If all users have certificates provisioned in Oracle Internet Directory, then you can set the `orclpkimatchingrule` to 1. This instructs Oracle Internet Directory to always conduct a certificate search. For instance, if your certificate authority does not support two common names in certificate DNs but the directory DNs are using two common names, then you would need to provision all user certificates into the directory. You can then set the `orclpkimatchingrule` to 1.

If you do not want to support the functionality of using different DNs in the PKI certificate and Oracle Internet Directory, then you can set the `orclpkimatchingrule` value to 0. You use this setting if all certificate DNs match directory DNs and you do not wish to provision the certificates.

You can also create your own mapping rules to map certificate DNs to directory DNs in Oracle Internet Directory 10g Release 2 (10.1.2.0.2). To use mapping rules, `orclpkimatchingrule` is set to 3 or 4.

When you want to use the mapping rule for all users, set `orclpkimatchingrule` to 3. If you also need to support certificate-based search and exact match, then set `orclpkimatchingrule` to 4.

Table 4-2 describes the values of the `orclpkimatchingrule` attribute.

Table 4-2 Oracle Internet Directory Matching Rules

Value	Rule
`orclpkimatchingrule=0`	Exact match. The bind is based on the subject DN of the client certificate. This DN is compared with the DN of the user in the directory.
`orclpkimatchingrule=1`	Certificate hash. The bind is based on the hashed value of the certificate.
`orclpkimatchingrule=2` (default)	Certificate hash/exact match. The bind is based on the hashed value of the certificate. If this operation fails, then a bind based on the subject DN of the client certificate is performed.
`orclpkimatchingrule=3`	Mapping rule only.
`orclpkimatchingrule=4`	Mapping rule/certificate hash/exact match. The bind is based on the mapping rule. If this operation fails, a bind based on the hashed value of the certificate is performed. If this operation fails, then a bind based on an exact match of the certificate is performed.

Note:

Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory and *Oracle Fusion Middleware Reference for Oracle Identity Management* for information on how to modify the `orclpkimatchingrule` attribute

> **Note:**
>
> A certificate search will fail if there is no user entry under the realm's user search base with that certificate, or if you are using an older version of Oracle Internet Directory that does not support the certificate search functionality. If the certificate search fails, then the database will revert to the old behavior of matching the user DN with the certificate DN for a successful connection.

- You have enabled SSL for your client-database Oracle Net connections as described in *Oracle Database Security Guide* for information about enabling SSL. Ensure that you included the following steps when you enabled SSL:
  - Enabled SSL for your database listener on `TCPS` and provided a corresponding TNS name
  - Stored your database PKI credentials in the database wallet that Database Configuration Assistant automatically created during database registration
- You have configured an SSL instance with two-way authentication for Oracle Internet Directory as described in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.
- You have prepared your directory by completing the tasks described in "Preparing the Directory for Enterprise User Security (Phase One)".
- You have configured your Enterprise User Security objects in the database and the directory by completing the tasks described in "Configuring Enterprise User Security Objects in the Database and the Directory (Phase Two)".

To configure Enterprise User Security for SSL authentication, perform the following tasks:

- Task 1: Enable the Enterprise Domain to Accept SSL Authentication
- Task 2: Set the LDAP_DIRECTORY_ACCESS Initialization Parameter to SSL
- Task 3: Connect as an SSL-Authenticated Enterprise User

Task 1: Enable the Enterprise Domain to Accept SSL Authentication

By default, OracleDefaultDomain is configured to accept all types of authentication. If this has been changed or if you are using another domain, then use Oracle Enterprise Manager to enable SSL authentication for your enterprise domain by performing the following steps:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.

6. Click **Manage Enterprise Domains**.

   The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.

7. Select **OracleDefaultDomain**. Click **Configure**.

   The Configure Domain page appears.

8. Click the **Configuration** tab.
9. Under User Authentication Types Accepted, select **SSL**.
10. Click **OK**.

Task 2: Set the LDAP_DIRECTORY_ACCESS Initialization Parameter to SSL

You can change this initialization parameter either by editing your database initialization parameter file or by issuing an `ALTER SYSTEM` SQL command with the `SET` clause.

For example, the following `ALTER SYSTEM` command changes the `LDAP_DIRECTORY_ACCESS` parameter value to `SSL` in the server parameter file:

```
ALTER SYSTEM SET LDAP_DIRECTORY_ACCESS=SSL SCOPE=SPFILE
```

> **See Also:**
>
> - *Oracle Database Reference* for information about editing initialization parameters
> - *Oracle Database Reference* for information about the `LDAP_DIRECTORY_ACCESS` initialization parameter
> - *Oracle Database SQL Language Reference* for information about using the `ALTER SYSTEM` command with the `SET` clause

Task 3: Connect as an SSL-Authenticated Enterprise User

Connecting as an SSL-authenticated enterprise user involves ensuring that you have the appropriate Oracle wallet features configured and that you do not have a wallet location specified in the client `sqlnet.ora` file. If the client `sqlnet.ora` file contains a wallet location, then multiple users and databases cannot share that file. Only the server `sqlnet.ora` file must have a value for the wallet location parameter.

To connect as an SSL-authentication enterprise user, perform the following steps:

1. Download the user wallet from the directory.
2. Use `orapki` utility to enable autologin for the user wallet. Enabling autologin generates a single sign-on (`.sso`) file and enables authentication to the SSL adapter.
3. Set the `TNS_ADMIN` environment variable (to point to the client's `sqlnet.ora` file) for the client if the client Oracle home points to a server Oracle home. (Because a server must have a wallet location set in its `sqlnet.ora` file and a client cannot have a wallet location specified there, the server and client cannot share `sqlnet.ora` files.)

   If you have a separate client Oracle home, then you do not need to set the `TNS_ADMIN` environment variable.

4. Launch SQL*Plus and enter the following at the command line:

   ```
   SQL> /@connect_identifier
   ```

   where `connect_identifer` is the Oracle Net service name you set up when you configured SSL for the database client.

   If your connection succeeds, then the system responds with `Connected to:...`. This is the confirmation message of a successful connect and setup. If an error message is displayed, then see "ORA-n Errors for SSL-Authenticated Enterprise Users".

   If you do connect successfully, then check that the appropriate global roles were retrieved from the directory, by entering the following at the SQL*Plus prompt:

   ```
   select * from session_roles
   ```

   If the global roles were not retrieved from the directory, then see "NO-GLOBAL-ROLES Checklist".

You have completed SSL-authenticated Enterprise User Security configuration.

> **Note:**
>
> For security purposes, ensure that you disable auto login for the user wallet after logging out from the enterprise user session with the database. This is especially important if the client computer is shared by more than one user.

This section includes the following topic: Viewing the Database DN in the Wallet and in the Directory.

> **See Also:**
>
> *Oracle Database Security Guide* for information about managing wallets with `orapki` utility.

4.5.3.1 Viewing the Database DN in the Wallet and in the Directory

When you use Database Configuration Assistant to register your database in the directory, this tool automatically creates identical DNs for the database wallet and the database directory entry. To view the database DN, use one of the following options:

Use Oracle Directory Manager to look in the directory under the realm Oracle Context for

```
cn=<short_database_name>,cn=OracleContext,<realm_DN>
```

where *short_database_name* is the first part of the fully qualified domain name for a database. For example, if you have a database named `db1.us.example.com`, then the short database name is `db1`.

- Use the following `mkstore` utility syntax on the command line:

  ```
  mkstore -wrl <wallet_location> -viewEntry ORACLE.SECURITY.DN
  ```

where *wallet_location* is the path to the database wallet.

> **See Also:**
>
> – "Troubleshooting Enterprise User Security" for information about diagnosing and resolving errors
>
> – Administering Enterprise User Security for information about configuring the identity management realm, and information about creating and managing enterprise domains, enterprise roles, and enterprise users

4.6 Troubleshooting Enterprise User Security

This section describes potential problems and associated corrective actions in the following topics:

- ORA-n Errors for Password-Authenticated Enterprise Users
- ORA-n Errors for Kerberos-Authenticated Enterprise Users
- ORA-n Errors for SSL-Authenticated Enterprise Users
- NO-GLOBAL-ROLES Checklist
- USER-SCHEMA ERROR Checklist
- DOMAIN-READ-ERROR Checklist

4.6.1 ORA-n Errors for Password-Authenticated Enterprise Users

If you receive an ORA-n error while using password-authenticated Enterprise User Security, then locate the error in the following section and take the recommended action.

ORA-1017: Invalid username/password; login denied
Cause: As in error message

Action: See "USER-SCHEMA ERROR Checklist"

ORA-28030: Server encountered problems accessing LDAP directory service
Cause: Indicates a problem with the connection between the database and the directory.

Action: Check the following:

1. Check that the correct `wallet_location` value is specified in the database's `sqlnet.ora` file in case you are not using the default wallet location. You can use Oracle Net Manager to enter the wallet location. You do not need to specify a wallet location in the `sqlnet.ora` file if the default wallet location is being used. If a wallet location is specified in the `sqlnet.ora` file, then you must ensure that it is correct.
2. If Domain Name System (DNS) server discovery of Oracle Internet Directory is not used, check that there is a correct `ldap.ora` file in `$LDAP_ADMIN`, `$ORACLE_HOME/ldap/admin`, `$TNS_ADMIN`, or `$ORACLE_HOME/network/admin`. (See *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about DNS server discovery.)

3. Check that the SSL port used (by way of either DNS discovery or an `ldap.ora` file) supports SSL with no authentication.
4. Check that the `LDAP_DIRECTORY_ACCESS` parameter is set to `PASSWORD` in the database initialization parameters file.
5. Use Database Configuration Assistant to reset the database password used to authenticate the database to Oracle Internet Directory. This resets it both locally in the database wallet, and remotely in the database entry in Oracle Internet Directory.
6. Check that the database wallet has autologin enabled. Either use `orapki` utility or check that there is a `cwallet.sso` file in `$ORACLE_HOME/admin/<ORACLE_SID>/wallet/`.
7. Use the password stored in the database wallet to check that the database can bind to Oracle Internet Directory:
   - Use the `mkstore` command-line utility to retrieve the database password from the wallet by using the following syntax:

     ```
     mkstore -wrl <database wallet location> -viewEntry
     ORACLE.SECURITY.PASSWORD
     ```

   - Use the password returned from `mkstore` in the following `ldapbind`:

     ```
     ldapbind -h <directory host> -p <non-SSL directory port> -D "<database
     DN>" -q
     Please enter bind password: Password returned by mkstore
     ```

8. Check to ensure that the database belongs to only one enterprise domain.

> **Note:**
>
> The `mkstore` utility is for troubleshooting purposes only. The name and functionality of this tool may change in the future.

ORA-28043: Invalid bind credentials for DB/OID connection
Cause: The database directory password no longer synchronizes with the directory.

Action: Use the **Regenerate Password** button in Database Configuration Assistant to generate a new directory password for the database, synchronize it with the directory, and store it in the database wallet.

ORA-28271: No permission to read user entry in LDAP directory service
Cause: As in error message

Action: Check the following:

1. Use Oracle Internet Directory Self-Service Console to check that a user search base containing this user is listed in the user search base attribute of the realm that you are using.
2. Check the ACL on the User Search Base in Oracle Internet Directory to ensure that the `verifierServices` group has read permission on the user entry, and that

this permission is not prevented by an ACL between the User Search Base entry and the user entry in the directory tree.

3. Check that the enterprise domain is in the password-accessible domains group for that realm Oracle Context.

ORA-28272: Domain policy restricts password-based GLOBAL user authentication.
Cause: As in error message

Action: Use the Oracle Enterprise Manager interface to set the user authentication policy for this enterprise domain to **Password** or **ALL**.

ORA-28273: No mapping for user nickname to LDAP distinguished name exists
Cause: As in error message

Action: Check the following:

1. Check that a user entry exists in Oracle Internet Directory for your user.
2. Use Oracle Internet Directory Self-Service Console to check that a user search base containing this user is listed in the identity management realm that you are using.
3. Check that the user entry contains the correct login name:
   - Use Oracle Internet Directory Self-Service Console to find the login name attribute that is configured for the directory in your realm, and
   - Check that the name provided during the attempted user database login is the value for that attribute in the user directory entry.
4. If you have an exclusive schema for the global user in the database, then check that the DN in the database matches the DN of the user entry in Oracle Internet Directory.

ORA-28274: No ORACLE password attribute corresponding to user nickname exists
Cause: As in error message

Action: Check the following:

1. Check that the user entry in the directory has the `orcluser` object class. If it does not, then perform the following steps:
   - Use Oracle Internet Directory Self-Service Console to check that the default object classes for new user creation include `orcluser`, and then
   - Use Oracle Internet Directory Self-Service Console to re-create the user, or
   - Add the `orcluser` and the `orcluserV2` object classes.
2. Check that there is a value for the attribute `orclpassword` in the user entry. If there is no value, then reset the user's directory password (`userpassword` attribute). This should prompt Oracle Internet Directory to regenerate the database password verifier for the user.
3. Use Oracle Internet Directory Self-Service Console to check that the user search base containing this user is listed in the user search base attribute of the realm that you are using.
4. Check that the ACL on the user search base attribute allows read and search access to the `orclpassword` attributes by the `verifierServices` group. This is set properly by default, but may have been altered.

ORA-28275: Multiple mappings for user nickname to LDAP distinguished name exist
Cause: There are multiple user DNs in the directory within the user search base whose login name for the user matches what was provided during the database connection.

Action: Use Oracle Internet Directory Self-Service Console to make the login name value unique (no two users share the same login name) within all user search bases associated with the realm Oracle Context.

ORA-28277: LDAP search, while authenticating global user with passwords, failed
Cause: As in error message

Action: Check that the relevant directory instance is up and running.

ORA-28278: No domain policy registered for password-based GLOBAL users
Cause: The database cannot read the enterprise domain information that it needs.

Action: See "DOMAIN-READ-ERROR Checklist"

ORA-28862: SSL connection failed
Cause: As in error message

Action: Check that you are using a non-SSL connect string.

4.6.2 ORA-n Errors for Kerberos-Authenticated Enterprise Users

If you receive an ORA-n error while using Kerberos-authenticated Enterprise User Security, then locate the error in the following section and take the recommended action.

ORA-1017: Invalid username/password; login denied
Cause: As in error message

Action: See "USER-SCHEMA ERROR Checklist"

ORA-28030: Problem accessing LDAP directory service
Cause: Indicates a problem with the connection between the database and the directory.

Action: See the actions listed for resolving ORA-28030: Server encountered problems accessing LDAP directory service in the troubleshooting section for password-authenticated enterprise users.

ORA-28271: No permission to read user entry in LDAP directory service
Cause: As in error message

Action: See the actions listed for resolving ORA-28271: No permission to read user entry in LDAP directory service in the troubleshooting section for password-authenticated enterprise users.

ORA-28292: No domain policy registered for Kerberos-based authentication
Cause: As in error message

Action: Perform the following actions:

1. Use Oracle Enterprise Manager to set the user authentication policy for this enterprise domain to **KERBEROS** or **ALL**.
2. See "DOMAIN-READ-ERROR Checklist"

ORA-28290: Multiple entries found for the same Kerberos principal name
Cause: The Kerberos principal name for this user is not unique within the user search base containing this user.

Action: Use Oracle Internet Directory Self-Service Console to change the Kerberos principal name, or to change the other copies so that it is unique.

ORA-28291: No Kerberos principal value found
Cause: As in error message

Action: Check the following:

1. Check that the user entry in the directory has the `krbprincipalname` attribute.

   If it does not have the `krbprincipalname` attribute, then check the following:

   - Check that the default attributes for new user creation by using Oracle Internet Directory Self-Service Console include `krbprincipalname`, and then
   - Use Oracle Internet Directory Self-Service Console to create the user again, or
   - Add the `orclcommonattributes` object class.
2. Check that there is a value for the attribute `krbprincipalname` in the user entry. If there is no value, then use Oracle Internet Directory Self-Service Console to enter one.
3. Use Oracle Internet Directory Self-Service Console to check that the user search base containing this user is listed in the realm Oracle Context that you are using.
4. Check that the ACL on the user search base attribute allows read and search access to the `krbprincipalname` attributes by the `verifierServices` group. This is set properly by default, but may have been altered.

ORA-28293: No matched Kerberos principal found in any user entry.
Cause: As in error message

Action: Check the following:

1. Check that a user entry exists in Oracle Internet Directory for your user.
2. Use Oracle Internet Directory Self-Service Console or `ldapsearch` to check that a user search base containing this user is listed in the identity management realm that you are using.
3. Check that the user entry in the directory contains the correct Kerberos principal name, by using the following steps:

   - Use Oracle Internet Directory Self-Service Console to find the Kerberos principal name attribute that is configured for the directory in your realm, and
   - Check that the correct Kerberos principal name appears in that attribute in the user's directory entry.
4. If you have an exclusive schema for the global user in the database, check that the DN in the database matches the DN of the user entry in Oracle Internet Directory.

ORA-28300: No permission to read user entry in LDAP directory service
Cause: As in error message

Action: Check that the database wallet contains the correct credentials for the database-to-directory connection. The wallet DN should be the DN of the database in Oracle Internet Directory. To retrieve the credentials, perform the following steps:

1. Use the `mkstore` command-line utility to retrieve the database password for the wallet by using the following syntax:

   ```
   mkstore -wrl <database wallet location> -viewEntry ORACLE.SECURITY.PASSWORD
   -viewEntry ORACLE.SECURITY.DN
   ```

2. If these values are incorrect, reset the database wallet by using Database Configuration Assistant.
3. Use the DN and the password returned by `mkstore` in the following `ldapbind`:

   ```
   ldapbind -h <directory host> -p <non-SSL directory port> -D "<database DN>"
   -q
   Please enter bind password: Password returned by mkstore
   ```

> **Note:**
>
> The `mkstore` utility is for troubleshooting purposes only. The name and functionality of this tool may change in the future.

ORA-28302: User does not exist in the LDAP directory service
Cause: As in error message

Action: Check that the user entry is present in the directory.

4.6.3 ORA-n Errors for SSL-Authenticated Enterprise Users

If you receive an ORA-n error while using SSL-authenticated Enterprise User Security, then locate the error in the following section and perform the recommended action.

ORA-1017: Invalid username/password; login denied
Cause: As in error message

Action: See "USER-SCHEMA ERROR Checklist"

ORA-28030: Problem accessing LDAP directory service
Cause: Indicates a problem with the connection between the database and the directory.

Action: Check the following:

1. Check that there is a correct `wallet_location` value in the database's `sqlnet.ora` file. If not, then use Oracle Net Manager to enter one.
2. If Domain Name System (DNS) server discovery of Oracle Internet Directory is not used, then check that there is a correct `ldap.ora` file in `$LDAP_ADMIN`, `$ORACLE_HOME/ldap/admin`, `$TNS_ADMIN` or `$ORACLE_HOME/`

network/admin. (See *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about DNS server discovery.)

3. Check that the Oracle Internet Directory SSL port used (by way of DNS discovery or an ldap.ora file) supports SSL with two-way authentication.
4. Check that the LDAP_DIRECTORY_ACCESS parameter is set to SSL in the database initialization parameters file.
5. Check that the database wallet has autologin enabled. Either use orapki utility or check that there is a cwallet.sso file in $ORACLE_HOME/admin/<ORACLE_SID>/wallet/.
6. Use the mkstore command-line utility to check that the database wallet has the database DN in it by using the following syntax:

   ```
   mkstore -wrl <database_wallet_location> -viewEntry ORACLE.SECURITY.DN
   ```

   If the wallet does not contain the database DN, then use Database Configuration Assistant to reregister the database with Oracle Internet Directory.
7. Check that the database can bind to Oracle Internet Directory, by using its wallet with the following ldapbind:

   ```
   ldapbind -h <directory_host> -p <directory_SSLport> -U 3 -W "file:<database
   wallet_location>" -Q
   Please enter SSL wallet password: wallet_password
   ```
8. Check to ensure that the database belongs to only one enterprise domain.

> **Note:**
>
> The mkstore utility is for troubleshooting purposes only. The name and functionality of this tool may change in the future.

ORA-28301: Domain policy has not been registered for SSL authentication
Cause: As in error message

Action: Use Oracle Enterprise Manager to set the user authentication policy for this enterprise domain to include SSL.

ORA-28862: SSL handshake failed
Cause: As in error message

Action: See the SSL (Secure Sockets Layer) chapter in *Oracle Database Security Guide* for information about configuring your SSL connection.

4.6.4 NO-GLOBAL-ROLES Checklist

If the enterprise user can connect to the database but a `select * from session_roles` returns no global roles, then check the following:

1. Check that the global role has been created in the database. To create global roles, use the following syntax:

   ```
   CREATE ROLE <role_name> IDENTIFIED GLOBALLY;
   ```
2. Use Oracle Enterprise Manager to check that the global role is included in an enterprise role in the directory.

3. Use Oracle Enterprise Manager to check that the enterprise role is assigned to the user in the directory.
4. If these checks are fine, then see the "DOMAIN-READ-ERROR Checklist".

4.6.5 USER-SCHEMA ERROR Checklist

If your database cannot read the user schema, then check the following:

1. If this is a globally authenticated administrative user who logged in as `SYSDBA` to shut down the database, the database shuts down successfully followed by the `ORA-01017: invalid username/password; logon denied` error, then ensure that the `sqlnet.ora` is configured with the wallet location of the Database-OID wallet. An Administrative user login and database shutdown and startup operation using this login does happen gracefully when the `sqlnet.ora` has the wallet location explicitly set to the directory where the Database-OID wallet file is present.
2. If this is an SSL-authenticated enterprise user, then ensure that the correct user wallet is being used by checking the following:
   - There is no `WALLET_LOCATION` parameter value in the client `sqlnet.ora` file, and
   - The `TNS_ADMIN` parameter is set properly so that the correct `sqlnet.ora` file is being used.
3. Check that the schema was created in the database as a global user, by using the following syntax:

   ```
   CREATE USER username IDENTIFIED GLOBALLY AS ' ';
   ```

   or by using the following syntax:

   ```
   CREATE USER username IDENTIFIED GLOBALLY AS '<DN>';
   ```

4. Suppose the following is true:
   - The user schema is an exclusive schema (created with the `CREATE USER` *`username`* `IDENTIFIED GLOBALLY AS '`*`user_DN`*`';` syntax), and
   - This is an SSL-authenticated user.

   Then, ensure that the DN in the user wallet matches the DN that was used in the `CREATE USER` statement.

   Use `orapki` to view the DN in the user wallet.

   Use the following syntax to view the DN that was used with the `CREATE USER` statement:

   ```
   SELECT EXTERNAL_NAME FROM DBA_USERS WHERE USERNAME='schema';
   ```

5. If you are using a shared schema, then check the following:
   - Use Oracle Enterprise Manager to ensure that you have created a user-schema mapping either for the entire enterprise domain or for the database.
   - If the user-schema mapping is intended to apply to this database (not to the entire enterprise domain), then check that the database can read its own entry and subtree in the directory.

     To check this, enter the following `ldapsearch` command for your database-to-directory connection type:

- If the database connects to the directory over SSL, then use

  ```
  ldapsearch -h directory_host -p directory_SSLport -U 3 -W
  "file:database_wallet_path" -Q -b "database_DN" "objectclass=*"
  Please enter SSL wallet password: wallet_password
  ```

  where *wallet_password* is the password to the wallet, which enables you to open or change the wallet.

- If the database connects to the directory by using password authentication, then use

  ```
  ldapsearch -h directory_host -p directory_port -D database_DN -q -b
  "database_DN" "objectclass=*"
  Please enter bind password: database_directory_password
  ```

  where *database_directory_password* is the database bind password returned by a utility like `mkstore`.

You should see the database entry and the relevant mapping.

- If the user-schema mapping applies to the entire enterprise domain rather than to only this individual database, then see "DOMAIN-READ-ERROR Checklist".

4.6.6 DOMAIN-READ-ERROR Checklist

If your database cannot read its enterprise domain information in Oracle Internet Directory, then check the following:

1. Use Oracle Enterprise Manager to check that the database is a member of exactly one enterprise domain, and add it to one if it is not.
2. Check that the database can see its domain, by entering one of the following at the command line:

   - If the database connects to the directory over SSL, then use

     ```
     ldapsearch -h directory_host -p directory_SSLport -U 3 -W
     "file:database_wallet_path" -Q -b "cn=OracleContext, realm_DN"
     "objectclass=orclDBEnterpriseDomain"
     Please enter SSL wallet password: wallet_password
     ```

     where *wallet_password* is the password to the wallet, which enables you to open or change the wallet.

   - If the database connects to the directory by using password authentication, then use

     ```
     ldapsearch -h directory_host -p directory_port -D database_DN -q -b
     "cn=OracleContext, realm_DN" "objectclass=orclDBEnterpriseDomain"
     Please enter bind password: database_directory_password
     ```

     where *database_directory_password* is the password in the database wallet, which is the database's password to Oracle Internet Directory.

   The `ldapsearch` command should return exactly one enterprise domain.

   If no domain is returned and Oracle Enterprise Manager shows the database as a member of a domain, then restart the database. Restarting the database updates the cached value for the enterprise domain.

   If more than one domain is returned, then use Oracle Enterprise Manager to remove the database from the additional domain.

3. Check that the database can read the enterprise domain subtree and thus can read its enterprise roles and mappings, by entering one of the following at the command line:

   - If the database connects to the directory over SSL, then use

     ```
     ldapsearch -h directory_host -p directory_SSLport -U 3 -W
     "file:database_wallet_path" -Q -b "cn=OracleContext, realm_DN"
     "objectclass=orclDBEnterpriseRole"
     Please enter SSL wallet password: wallet_password
     ```

     where *wallet_password* is the password to the wallet, which enables you to open or change the wallet.

   - If the database connects to the directory by using password authentication, then use

     ```
     ldapsearch -h directory_host -p directory_port -D database_DN -q -b
     "cn=OracleContext, realm_DN" "objectclass=orclDBEnterpriseRole"
     Please enter bind password: database_directory_password
     ```

     where *database_directory_password* is the password in the database wallet, which is the database password to Oracle Internet Directory.

   This `ldapsearch` should return all of the enterprise roles that you have created for this domain. If it does not, then use Oracle Enterprise Manager to create enterprise roles and mappings.

4. Use Oracle Enterprise Manager to set or reset the user authentication policy for the relevant enterprise domain. See "Configuring User Authentication Types" for information about setting the user authentication policy for an enterprise domain.

5

Administering Enterprise User Security

This chapter describes how to use Oracle Enterprise Manager to administer Enterprise User Security in Oracle Databases. This chapter contains the following topics:

- Administering Identity Management Realms
- Administering Enterprise Users
- Configuring User-Defined Enterprise Groups
- Configuring Databases for Enterprise User Security
- Administering Enterprise Domains

5.1 Administering Identity Management Realms

An identity management realm is a subtree of directory entries, all of which are governed by the same administrative policies. A realm Oracle Context is a subtree in a directory identity management realm that contains the data used by any installed Oracle product that uses the directory.

You can set properties of an identity management realm using Oracle Internet Directory tools like the Oracle Internet Directory Self-Service Console.

The Oracle Enterprise Manager Web interface enables you to manage Enterprise User Security related entries in an identity management realm.

This section describes administering identity management realms for Enterprise User Security. It contains the following topics:

- Identity Management Realm Versions
- Setting Properties of an Identity Management Realm
- Setting the Default Database-to-Directory Authentication Type for an Identity Management Realm
- Managing Identity Management Realm Administrators

> **Note:**
>
> Do not create users within a realm Oracle Context.

> **See Also:**
>
> - "How Oracle Internet Directory Implements Identity Management" for a discussion about identity management realms and realm Oracle Contexts and how they are related to one another
> - "About Enterprise User Security Directory Entries" for a discussion on the Oracle Internet Directory entries that are used for Enterprise User Security

5.1.1 Identity Management Realm Versions

Enterprise User Security can only use an identity management realm supplied by Oracle Internet Directory 10*g* (9.0.4), or later, which ships with Oracle Application Server 10*g* (9.0.4), or later.

You can manage Enterprise User Security directory entries in a version 9.0.4 (or later) identity management realm by using Oracle Enterprise Manager for Oracle Database.

5.1.2 Setting Properties of an Identity Management Realm

An identity management realm has a number of properties that can be viewed and managed by using Oracle Internet Directory tools like the Oracle Internet Directory Self-Service Console. These properties are described in Table 5-1.

Table 5-1 Identity Management Realm Properties

Property	Description
Attribute for Login Name	Name of the directory attribute used to store login names. By default, login names are stored in the `uid` attribute, but they can be changed to correspond to your directory configuration. In previous releases, this was the `cn` attribute.
Attribute for Kerberos Principal Name	Name of the directory attribute used to store Kerberos principal names. By default, Kerberos principal names are stored in the `krbPrincipalName` directory attribute, but they can be changed to correspond to your directory configuration by changing `orclCommonKrbPrincipalAttribute` in the identity management realm.
User Search Base	Full distinguished name (DN) for the node at which enterprise users are stored in the directory.
Group Search Base	Full DN for the node at which user groups are stored for this identity management realm in the directory.
Version Compatibility	This property is no longer used. However, you should ensure that it is not set to `81000`, because release 8.1.7 and earlier databases cannot be in the same realm with Oracle Database 10*g* or later databases.

> **Note:**
>
> Each identity management realm includes an `orcladmin` user who is the root user of that realm only. These realm-specific `orcladmin` users are represented by the directory entries `cn=orcladmin,cn=Users,`*`realm_DN`*. Note that when you are logged in to Enterprise User Security administration tools as a realm-specific `orcladmin` user, then you can manage only directory objects for that realm. To manage objects in another realm, you must log in to administration tools as the `orcladmin` user for that realm.
>
> This sections includes the following topic: Setting Login Name, Kerberos Principal Name, User Search Base, and Group Search Base Identity Management Realm Attributes.

5.1.2.1 Setting Login Name, Kerberos Principal Name, User Search Base, and Group Search Base Identity Management Realm Attributes

Setting these identity management realm attributes enables the database to locate Enterprise User Security entries.

To set Login Name, Kerberos Principal Name, User Search Base, and Group Search Base identity management realm attributes:

1. Log in to the Oracle Internet Directory Self-Service Console.

   Enter the URL to access the Oracle Internet Directory Self-Service Console in a browser window. For example:

   ```
   http://myhost1:7777/oiddas
   ```

   Log in as the `orcladmin` user.
2. Click the **Configuration** tab. Click the **Identity Management Realm** subtab.

   The Directory Configuration page appears.
3. Enter the appropriate information into the available fields.
4. Click **Submit** to save your changes to the directory.

> **See Also:**
>
> *Oracle Identity Management Guide to Delegated Administration* for detailed information on using the Oracle Internet Directory Self-Service Console

5.1.3 Setting the Default Database-to-Directory Authentication Type for an Identity Management Realm

The initial value for the `LDAP_DIRECTORY_ACCESS` parameter is picked from the default database-to-directory authentication attribute setting at the realm level. This parameter is set on individual databases when they are registered in Oracle Internet Directory.

The Oracle Enterprise Manager interface enables you to set the authentication mechanism that the database uses to authenticate to Oracle Internet Directory. The authentication mechanism can be set to password or SSL.

To set the default database-to-directory authentication type for an identity management realm:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **OID Realm Administration**.

   The OID Realm Administration page appears. The current DB-OID authentication method is displayed.
7. To change the current DB-OID authentication method, click **Change**.

   The Realm Configuration page appears.
8. Select Password or SSL under **DB-OID Authentication**.
9. Click **OK**.

5.1.4 Managing Identity Management Realm Administrators

An identity management realm contains administrative groups that have varying levels of privileges. The administrative groups for an identity management realm, which pertain to Enterprise User Security, are defined in Table 5-2. For more information about these groups, see "Administrative Groups".

Table 5-2 Enterprise User Security Identity Management Realm Administrators

Administrative Group	Definition
Oracle Database Registration Administrators (OracleDBCreators)	Registers new databases in the realm.
Oracle Database Security Administrators (OracleDBSecurityAdmins)	Has all privileges on the OracleDBSecurity directory subtree. Creates, modifies, and can read all Enterprise User Security directory objects.
Oracle Context Administrators (OracleContextAdmins)	Has full access to all groups and entries within its associated realm.
User Security Administrators (OracleUserSecurityAdmins)	Has relevant permissions necessary to administer security aspects for enterprise users in the directory. For example, OracleUserSecurityAdmins can modify user passwords.

To manage identity management realm administrators:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **OID Realm Administration**.

   The OID Realm Administration page appears. This page lists the Enterprise User Security related administrative groups in the identity management realm.
7. Select the administrative group that you wish to edit. Click **Edit**.

   The Edit page appears. It lists the directory users that are currently members of the group selected in the OID Realm Administration page.
8. To add a directory user to the group, click **Add**.

   The Select Users window appears.
9. Select the **Search Base**. The Search Base is the directory subtree that you wish to search for locating the user. Click **Go**.
10. Select the user that you wish to add as an administrator. Click **Select**.

    The user is added in the Edit page.
11. Click **OK**.

5.2 Administering Enterprise Users

This section describes how to use Oracle Internet Directory Self-Service Console and Oracle Enterprise Manager to administer enterprise users. It contains the following topics:

- Creating New Enterprise Users
- Setting Enterprise User Passwords
- Granting Enterprise Roles to Enterprise Users
- Granting Proxy Permissions to Enterprise Users
- Creating User-Schema Mappings for Enterprise Users
- Creating Label Authorizations for Enterprise Users

5.2.1 Creating New Enterprise Users

You can use Oracle Internet Directory tools like the Oracle Internet Directory Self-Service Console to create users in the directory.

> **Note:**
>
> Before creating new enterprise users, you must first define the user search base in the directory and also verify the user create base. See "Setting Login Name, Kerberos Principal Name, User Search Base, and Group Search Base Identity Management Realm Attributes"

To create new enterprise users:

1. Log in to the Oracle Internet Directory Self-Service Console.

   Enter the URL to access the Oracle Internet Directory Self-Service Console in a browser window. For example:

   ```
   http://myhost1:7777/oiddas
   ```

   Log in as the `orcladmin` user.
2. Click the **Directory** tab. Click the **Users** subtab.

   The Users page appears.
3. Click **Create** to create a new user.

   The Create User page appears.
4. Enter the appropriate user information in the Create User page. Click **Submit** to create a new enterprise user.

> **Note:**
>
> Note that if your users are authenticated to the database by using Kerberos credentials, and the `krbPrincipalName` attribute is not there, then see "Task 1: Configure Oracle Internet Directory Self-Service Console to display the Kerberos principal name attribute" for information about how to configure this.

5.2.2 Setting Enterprise User Passwords

You can use Oracle Internet Directory Self-Service Console to set and maintain enterprise user passwords in Oracle Internet Directory.

The enterprise user password is used for:

- Directory logon
- Database logon, to databases that support password authentication for global users

To set the password for an enterprise user:

1. Log in to the Oracle Internet Directory Self-Service Console.

   Enter the URL to access the Oracle Internet Directory Self-Service Console in a browser window. For example:

```
http://myhost1:7777/oiddas
```

Log in as the `orcladmin` user.

2. Click the **Directory** tab. Click the **Users** subtab.

   The Users page appears.

3. Enter part of the enterprise user's user name (login name) or e-mail address in the **Search** field. Click **Go**.

   A list of all users who match your search criteria displays.

4. Select the user for whom you wish to create a new password. Click **Edit**.

   The Edit User page appears.

5. Enter the new password in the **Password** field. Confirm the password in the **Confirm Password** field. Click **Submit**.

5.2.3 Granting Enterprise Roles to Enterprise Users

Enterprise roles are directory objects that allow you to group global roles from various databases. You can assign enterprise roles to enterprise users, which gives them privileges across enterprise databases.

To grant enterprise roles to enterprise users:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.

6. Click **Configure Enterprise Users**.

   The Configure Enterprise Users page appears.

7. Select the **Search Base** in which the enterprise user is located. The search base is the subtree which contains the enterprise user entry. You can optionally enter the common name of the enterprise user in the **Name** field. Select User in the **View** box. Click **Go**.

   A list of users with matching criteria appears.

8. Select the enterprise user that you wish to configure. Click **Configure**.

   The Configure User page appears.

9. Click the Enterprise Roles tab.
10. Click **Grant**.

    The Select Enterprise Roles window appears.

11. Select the enterprise role that you wish to grant. Click **Select**.
12. Click **OK** in the Configure User page.

5.2.4 Granting Proxy Permissions to Enterprise Users

Proxy permissions allow an enterprise user to proxy a local database user, which means that the enterprise user can log in to the database as the local database user. You can grant proxy permissions to individual users or groups. Proxy permissions are especially useful for middle tier applications that operate across multiple databases as enterprise users.

Proxy permissions are created at the enterprise domain level. After creating a proxy permission for an enterprise domain, you can grant it to an enterprise user.

To grant proxy permissions to enterprise users:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Configure Enterprise Users**.

   The Configure Enterprise Users page appears.
7. Select the **Search Base** in which the enterprise user is located. The search base is the subtree which contains the enterprise user entry. You can optionally enter the common name of the enterprise user in the **Name** field. Select User in the **View** box. Click **Go**.

   A list of users with matching criteria appears.
8. Select the enterprise user that you wish to configure. Click **Configure**.

   The Configure User page appears.
9. Click the **Proxy Permissions** tab.
10. Click **Grant**.

    The Select Proxy Permissions window appears.
11. Select the Proxy Permission to be granted. The proxy permission must have already been created for the enterprise domain. Click **Select**.
12. Click **OK** in the Configure User page.

5.2.5 Creating User-Schema Mappings for Enterprise Users

A user-schema mapping maps an enterprise user to a global database schema. When the enterprise user logs in to the database, he is connected to the mapped schema, by default.

To create a user-schema mapping:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.

2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Configure Enterprise Users**.

   The Configure Enterprise Users page appears.
7. Select the **Search Base** in which the enterprise user is located. The search base is the subtree which contains the enterprise user entry. You can optionally enter the common name of the enterprise user in the **Name** field. Select User in the **View** box. Click **Go**.

   A list of users with matching criteria appears.
8. Select the enterprise user that you wish to configure. Click **Configure**.

   The Configure User page appears.
9. Click the **User-Schema Mappings** tab. All user-schema maps that apply to the user directly or indirectly are displayed.

   A user can be individually mapped to a schema. Alternatively, you can map a directory subtree containing multiple users to the database schema.
10. Click **Create**.

    The Create Mapping page is displayed.
11. Under the From section, select **Users** to map an individual enterprise user to a database schema. Alternatively, select **Subtree** to map a directory subtree containing multiple users.
12. Under To, select **Database** to map to a database schema. Select **Domain** to map to a schema common to all databases in the enterprise domain.

    You can have multiple databases in an enterprise domain that have a common schema name. When you map an enterprise user to such a schema, the enterprise user is automatically mapped to the individual schemas in each database contained in the domain.
13. If you selected Database in the preceding step, then select the name of the database that contains the schema. Next, enter the database schema name. You can also use the search icon to select the schema. You will be required to log in to the database to select the schema.

    If you selected Domain in the preceding step, then select the name of the domain and enter the common schema name in the **Schema** field.
14. Click **Continue** in the Create Mapping page.
15. Click **OK** in the Configure User page.

5.2.6 Creating Label Authorizations for Enterprise Users

An Oracle Label Security (OLS) policy stored in the directory can have multiple profiles associated with it. Each profile is a set of policy authorizations and privileges. These policy

authorizations and privileges apply to all enterprise users who belong to the profile. You can assign a profile to an enterprise user.

To assign label authorizations to an enterprise user:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Configure Enterprise Users**.

   The Configure Enterprise Users page appears.
7. Select the **Search Base** in which the enterprise user is located. The search base is the subtree which contains the enterprise user entry. You can optionally enter the common name of the enterprise user in the **Name** field. Select User in the **View** box. Click **Go**.

   A list of users with matching criteria appears.
8. Select the enterprise user that you wish to configure. Click **Configure**.

   The Configure User page appears.
9. Click the **Label Authorizations** tab.

   A list of all user profiles associated with the user is displayed.
10. Click **Add**.

    The Select User Profile window appears.
11. Select the user profiles that you want the user to be added to, and click **Select**. You can only select one profile per policy.
12. Click **OK** in the Configure User page.

5.3 Configuring User-Defined Enterprise Groups

User-defined enterprise groups help group together enterprise users that require the same roles or privileges across enterprise databases. Enterprise groups are stored in the directory.

This section includes the following topic: Granting Enterprise Roles to User-Defined Enterprise Groups.

5.3.1 Granting Enterprise Roles to User-Defined Enterprise Groups

Enterprise roles are directory objects that allow you to group global roles from various databases. You can assign an enterprise role to an enterprise group, which gives the group members privileges across enterprise databases.

To grant an enterprise role to an enterprise group:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Configure User Defined Enterprise Groups**.

   The Configure Enterprise Groups page appears.
7. Select the **Search Base** in which the enterprise group is located. The search base is the subtree which contains the enterprise group entry. Optionally, enter the common name of the enterprise group in the **Name** field. Select Group in the **View** box. Click **Go**.

   A list of groups with matching criteria appears.
8. Select the enterprise group that you wish to configure. Click **Configure**.

   The Configure Group page appears.
9. Click the Enterprise Roles tab.

   A list of enterprise roles granted to the enterprise group is displayed.
10. Click **Grant** to grant a new enterprise role to the group.

    The Select Enterprise Roles window appears.
11. Select the enterprise roles that you wish to grant. Click **Select**.
12. Click **OK** in the Configure Group page.

5.4 Configuring Databases for Enterprise User Security

Enterprise User Security for databases registered with Oracle Internet Directory can be configured using Enterprise Manager. You can map users or subtrees to database schemas. You can also configure administrators in the directory that can modify schema mappings and enterprise domain membership of the database.

See the following sections for more information:

- Creating User-Schema Mappings for a Database
- Adding Administrators to Manage Database Schema Mappings

5.4.1 Creating User-Schema Mappings for a Database

A user-schema mapping maps an enterprise user to a global schema in the database. When the enterprise user logs in to the database, he is connected to the mapped schema, by default.

To create a user-schema mapping:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.

3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Configure Databases**.

   The Configure Databases page appears. A list of databases registered in the identity management realm is displayed.
7. Select the database name. Click **Configure**.

   The Configure Database page appears.
8. Click the **User-Schema Mappings** tab. All user-schema maps created at the database level are displayed. User-schema maps created at the enterprise domain levels are not displayed here.
9. Click **Create** to create a new user-schema mapping for the database.

   The Create Mapping page is displayed.
10. Under the From section, select **Users** to map an individual enterprise user to a database schema. Alternatively, select **Subtree** to map a directory subtree containing multiple users. You can use the Search icon to search for the appropriate user or subtree.
11. Under the To section, enter the name of the **Schema** to which the user or subtree should be mapped. You can use the search icon to search for the appropriate schema in the database. You will be required to log in to the database to access the schema names.
12. Click **Continue** in the Create Mapping page.
13. Click **OK** in the Configure Database page.

5.4.2 Adding Administrators to Manage Database Schema Mappings

Directory users who are authorized to manage database schema mappings for a database can create or delete database schema mappings for the database.

To add administrators for managing database schema mappings:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Configure Databases**.

The Configure Databases page appears. A list of databases registered in the identity management realm is displayed.

7. Select the database name. Click **Configure**.

   The Configure Database page appears.

8. Click the **Administrators** tab. A list of administrators who can manage database schema mappings is displayed.

9. Click **Add** to add an administrator.

   The Select Users window appears.

10. Select the **Search Base**. The Search Base is the directory subtree that you wish to search for locating the user. Click **Go**.

11. Select the user that you wish to add as an administrator. Click **Select**.

    The user is added in the Configure Database page.

12. If you want the user to be able to add or remove other administrators, then select the **Admin Group Owner** check box corresponding to the added user.

13. Click **OK**.

5.5 Administering Enterprise Domains

Enterprise Domains are groups of databases that can share enterprise roles, proxy permissions, user-schema mappings, and permitted authentication mechanisms. A database can belong to only one enterprise domain.

An enterprise domain can be thought of as an administrative domain, administered by the Domain Admins group for that domain. These administrators can add databases to the enterprise domain.

An identity management realm contains an enterprise domain called `OracleDefaultDomain`. `OracleDefaultDomain` is part of the realm when it is first created in the directory. When a new database is registered into a realm, it automatically becomes a member of `OracleDefaultDomain` in that realm. You can create and remove your own enterprise domains, but you must not remove `OracleDefaultDomain` from a realm.

This section describes how to use Oracle Enterprise Manager to administer enterprise domains in the directory. It contains the following topics:

- Creating an Enterprise Domain
- Adding Databases to an Enterprise Domain
- Creating User-Schema Mappings for an Enterprise Domain
- Configuring Enterprise Roles
- Configuring Proxy Permissions
- Configuring User Authentication Types
- Configuring Domain Administrators

5.5.1 Creating an Enterprise Domain

An enterprise domain is an administrative domain of databases that can share enterprise roles, proxy permissions, user-schema mappings, and permitted authentication mechanisms.

If you do not want to use `OracleDefaultDomain`, then you can create a new enterprise domain in your identity management realm.

To create an enterprise domain:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Manage Enterprise Domains**.

   The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.
7. Click **Create** to create a new enterprise domain.

   The Create Domain page appears.
8. Enter the name for the new enterprise domain in the **Name** field. Click **OK**.

   The new enterprise domain is added to the list of enterprise domains in the Enterprise Domains page.

5.5.2 Adding Databases to an Enterprise Domain

A member of the Domain Admins group can add databases to the enterprise domain. You can add databases to an enterprise domain from the Configure Domain page. You can also add databases from the Create Domain page, if you are creating a new enterprise domain.

> **Note:**
>
> The following restrictions apply to adding databases to an enterprise domain:
>
> - You can add a database to an enterprise domain only if both the database and the enterprise domain exist in the same realm.
> - A database cannot be added as a member of two different enterprise domains.

To add databases to an enterprise domain:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.

5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.

6. Click **Manage Enterprise Domains**.

   The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.

7. Select the enterprise domain that you wish to configure. Click **Configure**.

   The Configure Domain page appears.

8. Make sure that the Databases tab is selected. Click **Add** to add new databases to the enterprise domain.

   The Select Databases page appears. A list of databases, that are registered with the identity management realm, is displayed. You can add a database only if it is not part of any other enterprise domain.

9. Select the databases to add. Click **Select**.

10. Click **OK** in the Configure Domain page.

5.5.3 Creating User-Schema Mappings for an Enterprise Domain

A user-schema mapping maps an enterprise user to a global schema in the database. When the enterprise user logs in to the database, he is connected to the mapped schema, by default.

When you create a user-schema mapping for an enterprise domain, it applies to all databases in the domain. However, for the mapping to be effective in a database, that database must have a schema with the name used in the mapping.

To create a user-schema mapping for an enterprise domain:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.

6. Click **Manage Enterprise Domains**.

   The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.

7. Select the enterprise domain that you wish to configure. Click **Configure**.

   The Configure Domain page appears.

8. Click the **User-Schema Mappings** tab. All user-schema maps created at the domain level are displayed. User-schema maps created at database levels are not displayed here.
9. Click **Create** to create a new user-schema mapping for the domain.

The Create Mapping page is displayed.

10. Under the From section, select **Users** to map an individual enterprise user to a database schema. Alternatively, select **Subtree** to map a directory subtree containing multiple users. You can use the Search icon to search for the appropriate user or subtree.
11. Under the To section, enter the name of the **Schema** to which the user or subtree should be mapped.
12. Click **Continue** in the Create Mapping page.
13. Click **OK** in the Configure Domain page.

5.5.4 Configuring Enterprise Roles

An enterprise domain within an identity management realm can contain multiple enterprise role**s**. An enterprise role is a set of Oracle role-based authorization**s** across one or more databases in an enterprise domain.

Enterprise roles allow you to group global roles from different databases that are part of the enterprise domain. Enterprise roles can be assigned to enterprise users.

To create enterprise roles:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Manage Enterprise Domains**.

   The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.
7. Select the enterprise domain that you wish to configure. Click **Configure**.

   The Configure Domain page appears.
8. Click the **Enterprise Roles** tab.
9. Click **Create** to create a new enterprise role.

   The **Create Enterprise Role** page appears.
10. Enter a name for the enterprise role in the **Name** field. Click **Continue**.

    The new role is displayed in the Configure Domain page.

Next, you can configure the enterprise role that you just created. Configuring an enterprise role includes adding database global roles to the enterprise role and assigning the enterprise role to enterprise users or groups.

To add database global roles to the enterprise role:

1. Select the enterprise role that you just created in the Configure Domain page. Click **Edit**.

   The Edit Enterprise Role page is displayed.

2. Make sure that the **DB Global Roles** tab is selected. Click **Add** to add global roles from databases that are part of the enterprise domain.

   The Search and Select Database Global Roles page appears.

3. Select the **Database** that contains the global roles you wish to add. Log in to the selected database by supplying a **User Name** and **Password**. Click **Go**.

4. Select the global roles to add. Click **Select**.

   The selected roles appear in the Edit Enterprise Role page.

5. Repeat Steps 2 to 4 for the other databases.

You can now assign the enterprise role to enterprise users or groups.

To assign the enterprise role to enterprise users or groups:

1. Click the **Grantees** tab in the Edit Enterprise Role page.
2. Click **Add**.

   The Select Users or Groups page is displayed.

3. Select the **Search Base** or the subtree that contains the user or group. Select User under **View** if you are granting the enterprise role to a user. Select Group under **View**, if you are granting the role to a group. Optionally, enter the common name of the user or group in the **Name** field. Click **Go**.
4. Select the users or groups to be granted the enterprise role. Click **Select**.
5. Click **Continue** in the Edit Enterprise Role page.
6. Click **OK** in the Configure Domain page.

5.5.5 Configuring Proxy Permissions

Proxy permissions are created at the enterprise domain level. Proxy permissions allow an enterprise user to proxy a local database user, which means that the enterprise user can log in to the database as the local database user. You can grant proxy permissions to individual enterprise users or groups. Proxy permissions are especially useful for middle tier applications that operate across multiple databases as enterprise users.

To create a proxy permission for an enterprise domain:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.

6. Click **Manage Enterprise Domains**.

The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.

7. Select the enterprise domain that you wish to configure. Click **Configure**.

   The Configure Domain page appears.

8. Click the **Proxy Permissions** tab.

9. Click **Create** to create a new proxy permission.

   The **Create Proxy Permission** page appears.

10. Enter the name for the proxy permission in the **Name** field. Click **Continue**.

    The proxy permission appears in the Configure Domain page.

Next, you need to add target database users for the permission. You also need to grant the permission to enterprise users or groups, who can then proxy the target database users.

To add target database users for the proxy permission:

1. Select the proxy permission that you just created in the Configure Domain page. Click **Edit**.

   The Edit Proxy Permissions page appears.

2. Ensure that the **Target DB Users** tab is selected. Click **Add**.

   The Search and Select window appears. A list of all database users that have been altered to allow enterprise user proxy is displayed.

3. Select the target database users that you wish to proxy. Click **Select**.

You can now grant the proxy permission to enterprise users or groups.

To grant the proxy permission to an enterprise user or group:

1. Click the **Grantees** tab in the Edit Proxy Permission page.

2. Click **Add**.

   The Select Users or Groups window appears.

3. Select the **Search Base** or the subtree that contains the user or group. Select User under **View** if you are granting the proxy permission to a user. Select Group under **View**, if you are granting the proxy permission to a group. Optionally, enter the common name of the user or group in the **Name** field. Click **Go**.

4. Select the Users or Groups to grant them the proxy permission. Click **Select**.

5. Click **Continue** in the Edit Proxy Permission page.

6. Click **OK** in the Configure Domain page.

5.5.6 Configuring User Authentication Types

Enterprise users can be authenticated using password authentication, SSL authentication, or Kerberos authentication. You can set the authentication modes that are allowed for an enterprise domain using Enterprise Manager.

To configure user authentication types:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.

2. To navigate to your database, select **Databases** from the **Targets** menu.

3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Manage Enterprise Domains**.

   The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.
7. Select the enterprise domain that you wish to configure. Click **Configure**.

   The Configure Domain page appears.
8. Click the **Configuration** tab.
9. Under User Authentication Types Accepted, select the authentication types that you want to allow.
10. Click **OK**.

5.5.7 Configuring Domain Administrators

Domain administrators have full privileges in the domain. They can add or remove databases to the domain, create user-schema mappings, manage proxy permissions and modify domain configuration settings. You can add or remove domain administrators from Enterprise Manager.

To add an enterprise domain administrator:

1. Log in to Enterprise Manager Cloud Control, as an administrative user.
2. To navigate to your database, select **Databases** from the **Targets** menu.
3. Click the database name in the list that appears. The database page appears.
4. Under the **Administration** menu, select **Security, Enterprise User Security.** The Oracle Internet Directory Login page appears.
5. Enter the distinguished name (DN) of a directory user who can administer enterprise users in the **User** field. Enter the user password in the **Password** field. Click **Login**.

   The Enterprise User Security page appears.
6. Click **Manage Enterprise Domains**.

   The Manage Enterprise Domains page appears. This page lists the enterprise domains in the identity management realm.
7. Select the enterprise domain that you wish to configure. Click **Configure**.

   The Configure Domain page appears.
8. Click the **Administrators** tab. A list of administrators for the enterprise domain is displayed.
9. Click **Add** to add an administrator.

   The Select Users window appears.
10. Select the **Search Base**. The Search Base is the directory subtree that you wish to search for locating the user. Click **Go**.

11. Select the user that you wish to add as an administrator. Click **Select**.

    The user is added in the Configure Domain page.

12. If you want the user to be able to add or remove other administrators, then select the **Admin Group Owner** check box corresponding to the added user.

13. Click **OK**.

6

Enterprise User Security Manager (EUSM) Command Reference

Enterprise User Security Manager (EUSM) is a command-line tool you can use to manage the Enterprise User Security (EUS) Configuration in the Oracle Internet Directory (OID) directory server.

The EUSM command-line tool sends data to and retrieves data from the Oracle Internet Directory (OID) directory server. You can use Oracle Enterprise Manager to administer enterprise users, enterprise domains, and enterprise roles stored in OID as described in Oracle Enterprise Manager. However, this becomes a cumbersome process if the entries are very large and the process cannot be automated. Hence, the use of this command-line tool becomes a requirement.

The file path of the EUSM command is:

```
$ORACLE_HOME/bin
```

EUSM is a user friendly command-line tool. Entering `eusm` on the shell and pressing `Enter` or `Return`, prints all the commands that are supported. Also entering `eusm help <command>` or just `eusm <command>` and pressing `Enter` or `Return` prints the signature of a particular command supported by EUSM. Note that you must enter `eusm` in all lowercase characters.

Both EUSM commands and command-line options are not case sensitive.

Password credentials for the users: dbuser and ldap_user, and for the keystore can be stored in a client-side Oracle wallet in the external secure password store by providing information regarding the `<alias, username, password>` for each one. See About Using a Secure External Password Store, where this is described in more detail. Examples for each of the EUSM commands show this usage.

This chapter contains descriptions of the EUSM commands listed by their group. Each description contains the following parts:

Section	Description
Term	Describes the function of each term.
Syntax	Shows how to enter the command and provides a brief description of the basic uses of the command.
Options	Describes the function of each clause and option appearing in the syntax.
Usage Notes	Provides additional information on uses of the command and on how the command works.
Examples	Gives examples of the command using SSL port connectivity and not using SSL port connectivity to OID and using an Oracle wallet to store user credentials.

This chapter includes the following topics:

6.1 About Using a Secure External Password Store

If you want to use a secure external password store, then configure the Oracle wallet as described in the information that follows; otherwise, passwords can be provided interactively and you can skip this section.

Before you run the Enterprise User Security Manager (EUSM), configure a client-side Oracle wallet as a secure external password store so that your applications can use password credentials stored in the wallet to connect to databases. Storing database password credentials in a client-side Oracle wallet eliminates the need to embed passwords in application code, batch jobs, or scripts. This reduces the risk of exposing passwords in the clear in scripts and application code, and allows you to more easily manage password policies for user accounts without changing application code or scripts whenever passwords change.

See Configuring a Client to Use the External Password Store for steps to configure a client to use the external password store by using the `mkstore` command-line utility.

> **Note:**
>
> The external password store of the wallet is separate from the area where public key infrastructure (PKI) credentials are stored. Use the command-line utility `mkstore` to manage these credentials.

Using the `mkstore CreateCredential` command, configure the following user credentials by providing information for `<alias, username, password>`, in which you will be prompted to enter the password for each user.

- *dbalias*, dbuser, *password*
- *ldap_alias*, ldap_user, *password*
- *keystore_alias*, keystore, *passowrd*

Configuring these user credentials allows you to use the following parameters on the EUSM command line:

- dbalias=<db-password-alias>
- ldap_alias=<OID-user-password-alias>
- keystore_alias=<keystore-password-alias>

EUSM command examples use the following wallet credential information for users: `dbuser`, `ldap_user`, and `keystore` that was provided for the alias name, user name, and password. The wallet location is specified as shown.

- dbadmin1, sysman, *password*
- ldabadmin1, ldapman, *password*

- `keystore1, keystore, password`
- `wallet_location=/oracle/product/db_1/wallets`

EUSM command-line examples use the following entries on the command-line:

- `dbalias=dbadmin1`
- `ldap_alias=ldabadmin1`
- `keystore_alias=keystore1`
- `wallet_location=/oracle/product/db_1/wallets`

After configuring the client-side wallet, enable auto-login for Oracle Wallets to allow the administrator running EUSM commands to access and perform these services without having to supply the necessary credentials.

> **See Also:**
>
> - Managing the Secure External Password Store for Password Credentials for more information about creating a client-side password store wallet to store alias, user name, and password credentials for users

6.2 About SSL Port Connectivity through EUSM to OID

To enhance security, use the SSL Port connectivity option (`ldap_ssl_port=<OID ssl port>`) when connecting from Enterprise User Security Manager (EUSM) to Oracle Internet Directory (OID) directory server.

Using the SSL Port connectivity option (`ldap_ssl_port=<OID ssl port>`) assumes the environment where the OID directory server is set up supports the SSL port.

The following are additional parameters that must be given to EUSM to connect to the SSL port of the OID server:

- The `ldap_ssl_port` option takes the ssl port of the directory server (OID) as input from the EUSM command line.
- The `keystore=<path to PKCS12 format of keystore>` file path parameter takes the path to the PKCS12 format of the keystore (for example, `ewallet.p12` file) as input from the command line and the password is taken from the Oracle client-side wallet when the wallet is configured as a secure external password store or is taken interactively with the option `-K prompt for keystore password`.

Prerequisites

Prerequisites include the following:

- The client must have a keystore in PKCS12 format for example, ewallet.p12 file. This keystore file consists of a client private key certificate.
- The inputs for the passwords of keystore should also be given by the client.
- The client must have Java 2 SDK, v1.4 or any updated version that supports the current EUSM API.

6.3 Enterprise User Security Manager (EUSM) Command Summary

Enterprise User Security Manager (EUSM) commands are listed by group with links to its command page.

Group of Commands	Command	Description
Manage Enterprise Domains	listDomains	Lists the domains in the realm.
Manage Enterprise Domains	createDomain	Creates a domain in the realm.
Manage Enterprise Domains	deleteDomain	Deletes a domain from the realm.
Manage Enterprise Domains	listDomainInfo	Lists the domain information.
Manage Domain Administrators	addDomainAdmin	Adds a domain administrator.
Manage Domain Administrators	listDomainAdmins	Lists the domain administrators. The domain is taken as one of the inputs.
Manage Domain Administrators	removeDomainAdmin	Removes a domain administrator
Manage Databases in an Existing Domain	addDatabase	Adds a database to the domain.
Manage Databases in an Existing Domain	removeDatabase	Removes a database from the domain.
Manage Database Administrators	addDBAdmin	Adds a database administrator.
Manage Database Administrators	removeDBAdmin	Removes a database administrator.
Manage Database Administrators	listDBAdmins	Lists the database administrators.
Manage Database Administrators	listDBInfo	Lists the database information.
Manage user-schema mappings	createMapping	Creates the user and shared schema mapping.
Manage user-schema mappings	deleteMapping	Deletes a mapping.
Manage user-schema mappings	listMappings	Lists the user and shared schema mappings.
Setting Authentication Types	setAuthTypes	Sets authentication types to be accepted for the users in the domain.
Manage Enterprise Roles/ Global Roles	createRole	Creates an enterprise role.
Manage Enterprise Roles/ Global Roles	deleteRole	Deletes an enterprise role.
Manage Enterprise Roles/ Global Roles	addGlobalRole	Adds a global role or administrative role to an enterprise role.
Manage Enterprise Roles/ Global Roles	removeGlobalRole	Removes a global role or administrative role from an enterprise role.

Group of Commands	Command	Description
Manage Enterprise Roles/ Global Roles	grantRole	Grants an enterprise role.
Manage Enterprise Roles/ Global Roles	revokeRole	Revokes an enterprise role.
Manage Enterprise Roles/ Global Roles	listEnterpriseRoles	Lists the enterprise roles.
Manage Enterprise Roles/ Global Roles	listEnterpriseRolesOfUser	Lists the enterprise roles of a user.
Manage Enterprise Roles/ Global Roles	listEnterpriseRoleInfo	Lists enterprise role information.
Manage Enterprise Roles/ Global Roles	listGlobalRolesInDB	Lists the global roles in the database.
Manage Enterprise Roles/ Global Roles	listSharedSchemasInDB	Lists the shared schemas in the database.
Manage Proxy Authentication	createProxyPerm	Creates a proxy permission object.
Manage Proxy Authentication	deleteProxyPerm	Deletes a proxy permission object.
Manage Proxy Authentication	addTargetUser	Adds a target database user to the proxy permission object.
Manage Proxy Authentication	removeTargetUser	Removes a target database user from the proxy permission object.
Manage Proxy Authentication	grantProxyPerm	Maps an enterprise user to the database user through the proxy permission object.
Manage Proxy Authentication	revokeProxyPerm	Revokes a proxy permission object.
Manage Proxy Authentication	listProxyPermissions	Lists the proxy permissions. Input is the domain name.
Manage Proxy Authentication	listProxyPermissionsOfUser	Lists the proxy permissions for the user. Input is user distinguished name.
Manage Proxy Authentication	listProxyPermissionInfo	Lists the proxy permission information.
Manage Proxy Authentication	listTargetUsersInDB	Lists the target users in the database.
Manage Database-OID Authentication Method	setDBOIDAuth	Sets the database-OID authentication method.
Manage Database-OID Authentication Method	listDBOIDAuth	Lists the database-OID authentication method.
Manage the list of the Password Accessible Domains	addToPwdAccessibleDomains	Adds a domain to the password accessible domains group in the realm.
Manage the list of the Password Accessible Domains	removeFromPwdAccessibleDomains	Removes a domain from the password accessible domains group in the realm.
Manage the list of the Password Accessible Domains	listPwdAccessibleDomains	Lists the password accessible domains in the realm.
Display Realm Properties	listRealmCommonAttr	Lists the realm common attributes.
App Context Namespace	createAppCtxNamespace	Adds a new namespace.

Group of Commands	Command	Description
App Context Namespace	listAppCtxNamespaces	Lists the namespaces.
App Context Namespace	deleteAppCtxNamespace	Deletes a namespace.
App Context Attribute	createAppCtxAttribute	Adds a new attribute.
App Context Attribute	listAppCtxAttributes	Lists the attributes.
App Context Attribute	deleteAppCtxAttribute	Deletes an attribute.
App Context Attribute Value	createAppCtxAttributeValue	Adds a new attribute value.
App Context Attribute Value	listAppCtxAttributeValues	Lists the attribute values.
App Context Attribute Value	deleteAppCtxAttributeValue	Deletes an attribute value.
Manage App Context Users	createAppCtxUsers	Adds a new user for an attribute value.
Manage App Context Users	listAppCtxUsers	Lists all users for an attribute value.
Manage App Context Users	deleteAppCtxUsers	Deletes a user from an attribute value.
Help	help <*command name*>	Displays help for a command.

Examples of EUSM Commands Use Options

Examples for EUSM commands use some of the following options and values, where values are used for example purposes only:

- proxy_permission=PROXY01
- domain_name=test_domain
- domain_name=OracleDefaultDomain — an enterprise domain
- realm_dn=dc=yy, dc=company,dc=com
- user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
- database_name=dbtest1
- map_type=ENTRY — can be either ENTRY or SUBTREE
- map_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
- mapping_name=MAPPING01
- schema=test_user
- status=ENABLED — can be either ENABLED or DISABLED
- auth_type=SSL
- enterprise_role=ent_connect — enterprise role
- enterprise_role=ent_resource — global role
- global_role=global_resource
- global_role=global_connect
- dbuser=system — a privileged user
- db_alias=dbadmin1 — alias for dbuser credentials stored in an Oracle wallet
- dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
- target_user=PROXY_TEST

- namespace=ns1
- attribute_name=attr1
- attribute_value=val1
- ldap_host=xxxxx.zz.company.com — name of the OID server
- ldap_SSL_port=3131 — OID SSL (SASL) port used for OID connections; ports 3132 to 3141 or 13131 to 13141 can also be used
- keystore=/etc/myapp/keyStore — path to PKCS12 format of keystore; keystore location is administrator defined
- keystore_alias=keystore1 — alias for keystore credentials stored in an Oracle wallet
- ldap port =3060 — nonSSL (SASL) port used for OID connections; ports 3061 to 3070 or 13060 to 13070 can also be used
- ldap_user_dn=cn=orcladmin — OID administrator name
- ldap_alias=ldapadmin1 — alias for ldap_user credentials stored in an Oracle wallet
- wallet_location=/oracle/product/db_1/wallets — the wallet or secure external password store location

6.3.1 createDomain

Creates a domain in the realm.

Syntax

```
createDomain
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.

Option	Description
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include creating a domain in the realm with and without SSL port connectivity to OID.

Example 6-1 Creating a Domain in the Realm with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm createDomain domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-2 Creating a Domain in the Realm with SSL Port Conectivity to OID

```
eusm createDomain domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-3 Creating a Domain in the Realm with non-SSL Port Conectivity to OID

```
eusm createDomain domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.2 deleteDomain

Deletes a domain from the realm.

Syntax

```
deleteDomain
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
```

```
ldap_user_dn=<DN of OID user>
ldap_alias=<OID user password alias>
wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_port=<OID non ssl port>`	OID non ssl port.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include deleting a domain in the realm with and without SSL port connectivity to OID.

Example 6-4 Deleting a Domain from the Realm with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm deleteDomain domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-5 Deleting a Domain from the Realm with SSL Port Conectivity to OID

```
eusm deleteDomain domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-6 Deleting a Domain from the Realm with non-SSL Port Conectivity to OID

```
eusm deleteDomain domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.3 listDomains

Lists the domains in the realm.

Syntax

```
listDomains
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include listing the domains in the realm with and without SSL port connectivity to OID.

Example 6-7 Lists the domains in the realm with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listDomains realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-8 Lists the domains in the realm with SSL Port Conectivity to OID

```
eusm listDomains realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-9 Lists the domains in the realm with non-SSL Port Conectivity to OID

```
eusm listDomains realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.4 listDomainInfo

List domain information.

Syntax

```
listDomainInfo
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.

Option	Description
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include listing the domain information with and without SSL port connectivity to OID.

Example 6-10 Listing the Domain Information with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listDomainInfo domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-11 Listing the Domain Information with SSL Port Conectivity to OID

```
eusm listDomainInfo domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-12 Listing the Domain Information with non-SSL Port Conectivity to OID

```
eusm listDomainInfo domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.5 addDomainAdmin

Adds a domain administrator.

Syntax

```
addDomainAdmin
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     user_dn=<user DN>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
```

```
keystore_alias=<keystore password alias>
ldap_user_dn=<DN of OID user>
ldap_alias=<OID user password alias>
wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`user_dn=<user DN>`	DN of the user. For example, the user to be added as database administrator in the command addDBAdmin.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include adding a domain administrator with and without SSL port connectivity to OID.

Example 6-13 Adding a Domain Administrator with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm addDomainAdmin domain_name=test_domain
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com realm_dn=dc=yy,
dc=company,dc=com ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131
keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/oracle/
product/db_1/wallets
```

Example 6-14 Adding a Domain Administrator with SSL Port Conectivity to OID

```
eusm addDomainAdmin domain_name=test_domain
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com realm_dn=dc=yy,
```

```
dc=company,dc=com ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131
keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-15 Adding a Domain Administrator with non-SSL Port Conectivity to OID

```
eusm addDomainAdmin domain_name=test_domain
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com realm_dn=dc=yy,
dc=company,dc=com ldap_host=xxxxx.zz.company.com ldap_port=3060
ldap_user_dn=cn=orcladmin
```

6.3.6 removeDomainAdmin

Removes a domain administrator.

Syntax

```
removeDomainAdmin
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     user_dn=<user DN>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`user_dn=<user DN>`	DN of the user. For example, the user to be removed as database administrator in the command removeDBAdmin.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.

Option	Description
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include removing a domain administrator with and without SSL port connectivity to OID.

Example 6-16 Removing a Domain Administrator with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm removeDomainAdmin domain_name=test_domain
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com realm_dn=dc=yy,
dc=company,dc=com ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131
keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/oracle/
product/db_1/wallets
```

Example 6-17 Removing a Domain Administrator with SSL Port Conectivity to OID

```
eusm removeDomainAdmin domain_name=test_domain
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com realm_dn=dc=yy,
dc=company,dc=com ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131
keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-18 Removing a Domain Administrator with non-SSL Port Conectivity to OID

```
eusm removeDomainAdmin domain_name=test_domain
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com realm_dn=dc=yy,
dc=company,dc=com ldap_host=xxxxx.zz.company.com ldap_port=3060
ldap_user_dn=cn=orcladmin
```

6.3.7 listDomainAdmins

Lists the domain administrators. The domain is taken as one of the inputs.

Syntax

```
listDomainAdmins
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
```

```
keystore_alias=<keystore password alias>
ldap_user_dn=<DN of OID user>
ldap_alias=<OID user password alias>
wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include listing domain administrators with and without SSL port connectivity to OID.

Example 6-19 Listing the Domain Administrators with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listDomainAdmins domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-20 Listing the Domain Administrators with SSL Port Conectivity to OID

```
eusm listDomainAdmins domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-21 Listing the Domain Administrators with non-SSL Port Conectivity to OID

```
eusm listDomainAdmins domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.8 addDatabase

Adds a database to the domain.

Syntax

```
addDatabase
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     database_name=<Database name>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`database_name=<Database name>`	Database name.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port>	ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include adding a database to the domain with and without SSL port connectivity to OID.

Example 6-22 Adding a Database to the Domain with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm addDatabase domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-23 Adding a Database to the Domain with SSL Port Conectivity to OID

```
eusm addDatabase domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-24 Adding a Database to the Domain with non-SSL Port Conectivity to OID

```
eusm addDatabase domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.9 removeDatabase

Removes a database from the domain.

Syntax

```
removeDatabase
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     database_name=<Database name>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
domain_name=<domain name>	Name of the domain.
realm_dn=<DN of the realm>	DN of the realm.
database_name=<Database name>	Database name.
ldap_host=<OID host>	OID host.
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>	OID non ssl port or OID ssl port.
keystore=<path to keystore>	Path to the keystore.
keystore_alias=<keystore password alias>	Keystore password from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include removing a database from the domain with and without SSL port connectivity to OID.

Example 6-25 Removing a Database from the Domain with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm removeDatabase domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131
keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/oracle/
product/db_1/wallets
```

Example 6-26 Removing a Database from the Domain with SSL Port Conectivity to OID

```
eusm removeDatabase domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131
keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-27 Removing a Database from the Domain with non-SSL Port Conectivity to OID

```
eusm removeDatabase domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 ldap_host=xxxxx.zz.company.com ldap_port=3060
ldap_user_dn=cn=orcladmin
```

6.3.10 addDBAdmin

Adds a database administrator.

Syntax

```
addDBAdmin
     realm_dn=<DN of the realm>
     database_name=<Database name>
     user_dn=<Distinguished name of the user>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`realm_dn=<DN of the realm>`	DN of the realm.
`database_name=<Database name>`	Database name.
`user_dn=<user DN>`	DN of the user. For example, the user to be added as database administrator in the command addDBAdmin.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include adding a database administrator with and without SSL port connectivity to OID.

Example 6-28 Adding a Database Administrator with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm addDBAdmin realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-29 Adding a Database Administrator with SSL Port Conectivity to OID

```
eusm addDBAdmin realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-30 Adding a Database Administrator with non-SSL Port Conectivity to OID

```
eusm addDBAdmin realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.11 listDBAdmins

Lists the database administrators.

Syntax

```
listDBAdmins
     realm_dn=<DN of the realm>
     database_name=<Database name>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`realm_dn=<DN of the realm>`	DN of the realm.
`database_name=<Database name>`	Database name.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port>	ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.

Option	Description
keystore=<path to keystore>	Path to the keystore.
keystore_alias=<keystore password alias>	Keystore password taken from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include listing the database administrators with and without SSL port connectivity to OID.

Example 6-31 Listing the Database Administrators with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listDBAdmins realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-32 Listing the Database Administrators with SSL Port Conectivity to OID

```
eusm listDBAdmins realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-33 Listing the Database Administrators with non-SSL Port Conectivity to OID

```
eusm listDBAdmins realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 ldap_host=xxxxx.zz.company.com ldap_port=3060
ldap_user_dn=cn=orcladmin
```

6.3.12 listDBInfo

Lists the database information.

Syntax

```
listDBInfo
     realm_dn=<DN of the realm>
```

```
database_name=<Database name>
ldap_host=<OID host>
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
keystore=<path to keystore>
keystore_alias=<keystore password alias>
ldap_user_dn=<DN of OID user>
ldap_alias=<OID user password alias>
wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`realm_dn=<DN of the realm>`	DN of the realm.
`database_name=<Database name>`	Database name.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port>	ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include listing the database information with and without SSL port connectivity to OID.

Example 6-34 Lists the Database Information with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listDBInfo realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-35 Lists the Database Information with SSL Port Conectivity to OID

```
eusm listDBInfo realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-36 Lists the Database Information with non-SSL Port Conectivity to OID

```
eusm listDBInfo realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.13 removeDBAdmin

Removes a database administrator.

Syntax

```
removeDBAdmin
     realm_dn=<DN of the realm>
     database_name=<Database name>
     user_dn=<Distinguished name of the user>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`realm_dn=<DN of the realm>`	DN of the realm.
`database_name=<Database name>`	Database name.
`user_dn=<user DN>`	DN of the user. For example, the user to be added as database administrator in the command addDBAdmin.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include removing a database administrator with and without SSL port connectivity to OID.

Example 6-37 Removing a Database Administrator with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm removeDBAdmin realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-38 Removing a Database Administrator with SSL Port Conectivity to OID

```
eusm removeDBAdmin realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-39 Removing a Database Administrator with non-SSL Port Conectivity to OID

```
eusm removeDBAdmin realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.14 createMapping

Creates the user and shared schema mapping.

Syntax

```
createMapping
    [domain_name=<domain name>]
    [database_name=<database name>]
     realm_dn=<DN of the realm>
     map_type=<mapping type ENTRY/SUBTREE>
     map_dn=<DN which is being mapped to schema>
     schema=<database schema>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`[domain_name=<domain name>]`	Name of the domain.
`[database_name=<database name>]`	Database name.
`realm_dn=<DN of the realm>`	DN of the realm.
`map_type=<mapping type ENTRY/ SUBTREE>`	Type of mapping ENTRY/SUBTREE.
`map_dn=<DN which is being mapped to schema>`	DN that is being mapped to the schema.
`schema=<database schema>`	Database schema.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include creating the user or shared schema mapping with and without SSL port connectivity to OID.

Example 6-40 Creating the User or Shared Schema Mapping with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm createMapping database_name=dbtest1
realm_dn=dc=yy,dc=company,dc=com map_type=ENTRY
map_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com schema=test_user
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-41 Creating the User or Shared Schema Mapping with SSL Port Conectivity to OID

```
eusm createMapping database_name=dbtest1 realm_dn=dc=yy,dc=company,dc=com
map_type=ENTRY map_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
schema=test_user ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131
keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-42 Creating the User or Shared Schema Mapping with non-SSL Port Conectivity to OID

```
eusm createMapping database_name=dbtest1 realm_dn=dc=yy,dc=company,dc=com
map_type=ENTRY map_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
schema=test_user ldap_host=xxxxx.zz.company.com ldap_port=3060
ldap_user_dn=cn=orcladmin
```

6.3.15 deleteMapping

Deletes a mapping.

Syntax

```
deleteMapping
    [domain_name=<domain name>]
    [database_name=<database name>]
     realm_dn=<DN of the realm>
     mapping_name=<Name of mapping>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`[domain_name=<domain name>]`	Name of the domain.
`[database_name=<database name>]`	Database name.
`realm_dn=<DN of the realm>`	DN of the realm.
`mapping_name=<Name of mapping>`	Name of the mapping.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.

Option	Description
keystore_alias=<keystore password alias>	Keystore password taken from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include deleting the user or shared schema mapping with and without SSL port connectivity to OID.

Example 6-43 Deleting the User or Shared Schema Mapping with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm deleteMapping database_name=dbtest1
realm_dn=dc=yy,dc=company,dc=com mapping_name=MAPPING01
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-44 Deleting the User or Shared Schema Mapping with SSL Port Conectivity to OID

```
eusm deleteMapping database_name=dbtest1
realm_dn=dc=yy,dc=company,dc=com mapping_name=MAPPING01
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-45 Deleting the User or Shared Schema Mapping with non-SSL Port Conectivity to OID

```
eusm deleteMapping database_name=dbtest1
realm_dn=dc=yy,dc=company,dc=com mapping_name=MAPPING01
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.16 listMappings

Lists the user and shared schema mappings.

Prerequisites

(Optional) List the prerequisites for executing the command in the following list:

- Prerequisite #1
- Prerequisite #2

Syntax

```
listMappings
    [domain_name=<domain name>]
    [database_name=<database name>]
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`[domain_name=<domain name>]`	Name of the domain.
`[database_name=<database name>]`	Database name.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include listing the user or shared schema mappings with and without SSL port connectivity to OID.

Example 6-46 Listing the User or Shared Schema Mappings with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listMappings database_name=dbtest1 realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-47 Listing the User or Shared Schema Mappings with SSL Port Conectivity to OID

```
eusm listMappings database_name=dbtest1
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-48 Listing the User or Shared Schema Mappings with non-SSL Port Conectivity to OID

```
eusm listMappings database_name=dbtest1
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.17 setAuthTypes

Sets the authentication types to be accepted for the users in the domain

Syntax

```
setAuthTypes
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     auth_types=<Allowed User-DB authentication>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`auth_types=<Allowed User-DB authentication>`	Allowed user-database authentication types
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.

Option	Description
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include setting the authentication type accepted for user in the domain with and without SSL port connectivity to OID.

Example 6-49 Setting the Authentication Types Accepted for the Users in the Domain with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm setAuthTypes domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
auth_type=SSL ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131
keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/oracle/
product/db_1/wallets
```

Example 6-50 Setting the Authentication Types Accepted for the Users in the Domain with SSL Port Conectivity to OID

```
eusm setAuthTypes domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
auth_type=SSL ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131
keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-51 Setting the Authentication Types Accepted for the Users in the Domain with non-SSL Port Conectivity to OID

```
eusm setAuthTypes domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
auth_type=SSL ldap_host=xxxxx.zz.company.com ldap_port=3060
ldap_user_dn=cn=orcladmin
```

6.3.18 createRole

Creates an enterprise role.

Syntax

```
createRole
     enterprise_role=<Enterprise role name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
```

```
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
keystore=<path to keystore>
keystore_alias=<keystore password alias>
ldap_user_dn=<DN of OID user>
ldap_alias=<OID user password alias>
wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`enterprise_role=<Enterprise role name>`	Name of the enterprise role.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Enterprise roles are access privileges assigned to enterprise users. Enterprise roles are a set of Oracle role-based authorizations across one or more databases in an enterprise domain. Enterprise roles are stored in the directory and contain one or more global roles.

A global role is a role managed in a directory, but its privileges are contained within a single database. A global role is created in a database using SQL*Plus. For example:

```
SQL> create role global_connect identified globally;
SQL> create role global_resource identified globally;
SQL> grant connect to global_connect;
SQL> grant resource to global_resource;
```

Examples

Examples include creating an enterprise role in an enterprise domain in the realm with and without SSL port connectivity to OID.

Example 6-52 Creating a Role with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm createRole enterprise_role=ent_connect domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/oracle/
product/db_1/wallets
```

Example 6-53 Creating a Role with SSL Port Conectivity to OID

```
eusm createRole enterprise_role=ent_connect domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-54 Creating a Role with non-SSL Port Conectivity to OID

```
eusm createRole enterprise_role=ent_connect domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.19 deleteRole

Deletes an enterprise role.

Syntax

```
deleteRole
     enterprise_role=<Enterprise role name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`enterprise_role=<Enterprise role name>`	Name of the enterprise role.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.

Option	Description
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Enterprise roles are access privileges assigned to enterprise users. Enterprise roles are a set of Oracle role-based authorizations across one or more databases in an enterprise domain. Enterprise roles are stored in the directory and contain one or more global roles.

A global role is a role managed in a directory, but its privileges are contained within a single database. A global role is created in a database using SQL*Plus. For example:

```
SQL> create role global_connect identified globally;
SQL> create role global_resource identified globally;
SQL> grant connect to global_connect;
SQL> grant resource to global_resource;
```

Examples

Examples include deleting an enterprise role in an enterprise domain in the realm with and without SSL port connectivity to OID.

Example 6-55 Deleting a Role with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm deleteRole enterprise_role=ent_connect
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-56 Deleting a Role with SSL Port Conectivity to OID

```
eusm deleteRole enterprise_role=ent_connect
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-57 Deleting a Role with non-SSL Port Conectivity to OID

```
eusm deleteRole enterprise_role=ent_connect domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.20 addGlobalRole

Adds a global role or an administrative role to an enterprise role.

Syntax

```
addGlobalRole
     enterprise_role=<Enterprise role name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     database_name=<Database name>
     global_role=<Global role name>
     dbuser=<Database user name to connect>
     db_alias=<Database username password alias>
     dbconnect_string=<Database connect string>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`enterprise_role=<Enterprise role name>`	Name of the enterprise role.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`database_name=<Database name>`	Database name.
`global_role=<Global role name>`	Global role or administrative role name.
`dbuser=<Database user name to connect>`	The database user name to connect.
`db_alias=<Database username password alias>`	Database user password taken from the Oracle wallet.
`dbconnect_string=<Database connect string>`	Database connect string
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port>	ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.

Option	Description
keystore_alias=<keystore password alias>	Keystore password taken from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

Enterprise roles are access privileges assigned to enterprise users. Enterprise roles are a set of Oracle role-based authorizations across one or more databases in an enterprise domain. Enterprise roles are stored in the directory and contain one or more global roles.

A global role and an administrative role are roles managed in a directory, but their privileges are contained within a single database. A global role and an administrative role are created in a database using SQL*Plus. A `global_role` for administrative role can be either `SYSDBA`, `SYSOPER`, `SYSBACKUP`, `SYSKM`, or `SYSDG`. For example:

```
SQL> create role global_connect identified globally;
SQL> create role global_resource identified globally;
SQL> grant connect to global_connect;
SQL> grant resource to global_resource;
```

Examples

Examples include adding a global role and an administrative role in an enterprise domain in the realm for a database user with and without SSL port connectivity to OID.

Example 6-58 Adding a Global Role with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm addGlobalRole enterprise_role=ent_resource
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 global_role=global_resource dbuser=system
db_alias=dbadmin1 dbconnect_string=zzzz10-
yy.yy.company.com:1531:dbtest1  ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-59 Adding an Administrative Role with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm addGlobalRole enterprise_role=ent_resource
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 global_role=SYSDBA dbuser=system
db_alias=dbadmin1 dbconnect_string=zzzz10-
yy.yy.company.com:1531:dbtest1 ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore
```

```
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-60 Adding a Global Role with SSL Port Conectivity to OID

```
eusm addGlobalRole enterprise_role=ent_resource
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 global_role=global_resource dbuser=system
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-61 Adding an Administrative Role with SSL Port Conectivity to OID

```
eusm addGlobalRole enterprise_role=ent_resource
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 global_role=SYSDBA dbuser=system
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-62 Adding a Global Role with non-SSL Port Conectivity to OID

```
eusm addGlobalRole enterprise_role=ent_resource
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 global_role=global_resource dbuser=system
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

Example 6-63 Adding an Administrative Role with non-SSL Port Conectivity to OID

```
eusm addGlobalRole enterprise_role=ent_resource
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 global_role=SYSDBA dbuser=system
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.21 removeGlobalRole

Removes a global role or an administrative role from an enterprise role.

Syntax

```
removeGlobalRole
     enterprise_role=<Enterprise role name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     database_name=<Database name>
     global_role=<Global role name >
     dbuser=<Database user name to connect>
     db_alias=<Database username password alias>
```

```
dbconnect_string=<Database connect string>
ldap_host=<OID host>
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
keystore=<path to keystore>
keystore_alias=<keystore password alias>
ldap_user_dn=<DN of OID user>
ldap_alias=<OID user password alias>
wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`enterprise_role=<Enterprise role name>`	Name of the enterprise role.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`database_name=<Database name>`	Database name.
`global_role=<Global role name>`	Global role or administrative role name.
`dbuser=<Database user name to connect>`	The database user name to connect.
`db_alias=<Database username password alias>`	Database user password taken from the Oracle wallet.
`dbconnect_string=<Database connect string>`	Database connect string
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Enterprise roles are access privileges assigned to enterprise users. Enterprise roles are a set of Oracle role-based authorizations across one or more databases in an enterprise domain. Enterprise roles are stored in the directory and contain one or more global roles.

A global role and an administrative role are roles managed in a directory, but their privileges are contained within a single database. A global role and an administrative

role are created in a database using SQL*Plus. A `global_role` for administrative role can be either `SYSDBA`, `SYSOPER`, `SYSBACKUP`, `SYSKM`, or `SYSDG`. For example:

```
SQL> create role global_connect identified globally;
SQL> create role global_resource identified globally;
SQL> grant connect to global_connect;
SQL> grant resource to global_resource;
```

Examples

Examples include removing a global role and an administrative role in an enterprise domain in the realm from a database user with and without SSL port connectivity to OID.

Example 6-64 Removing a Global Role with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm removeGlobalRole enterprise_role=ent_resource
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 global_role=global_connect dbuser=system
db_alias=dbadmin1 dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-65 Removing an Administrative Role with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm removeGlobalRole enterprise_role=ent_resource
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 global_role=SYSDBA dbuser=system db_alias=dbadmin1
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-66 Removing a Global Role with SSL Port Conectivity to OID

```
eusm removeGlobalRole enterprise_role=ent_resource
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 global_role=global_connect dbuser=system
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-67 Removing an Administrative Role with SSL Port Conectivity to OID

```
eusm removeGlobalRole enterprise_role=ent_resource
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 global_role=SYSDBA dbuser=system
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-68 Removing a Global Role with non-SSL Port Conectivity to OID

```
eusm removeGlobalRole enterprise_role=ent_resource
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 global_role=global_connect dbuser=system
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

Example 6-69 Removing an Administrative Role with non-SSL Port Conectivity to OID

```
eusm removeGlobalRole enterprise_role=ent_resource
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 global_role=SYSDBA dbuser=system
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.22 grantRole

Grants an enterprise role.

Syntax

```
grantRole
     enterprise_role=<Enterprise role name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     [user_dn=<Distinguished name of user>]
     [group_dn=<Distinguished name of group>]
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`enterprise_role=<Enterprise role name>`	Name of the enterprise role.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`[user_dn=<Distinguished name of user>]`	DN of the user.
`[group_dn=<Distinguished name of group>]`	DN of the group.

Option	Description
ldap_host=<OID host>	OID host.
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>	OID non ssl port or OID ssl port.
keystore=<path to keystore>	Path to the keystore.
keystore_alias=<keystore password alias>	Keystore password taken from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

Enterprise roles are access privileges assigned to enterprise users. Enterprise roles are a set of Oracle role-based authorizations across one or more databases in an enterprise domain. Enterprise roles are stored in the directory and contain one or more global roles.

A global role is a role managed in a directory, but its privileges are contained within a single database. A global role is created in a database using SQL*Plus. For example:

```
SQL> create role global_connect identified globally;
SQL> create role global_resource identified globally;
SQL> grant connect to global_connect;
SQL> grant resource to global_resource;
```

Examples

Examples include granting an enterprise role in an enterprise domain in the realm to a user with and without SSL port connectivity to OID.

Example 6-70 Granting a Role to a User with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm grantRole enterprise_role=ent_connect domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-71 Granting a Role to a User with SSL Port Conectivity to OID

```
eusm grantRole enterprise_role=ent_connect domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-72 Granting a Role to a User with non-SSL Port Conectivity to OID

```
eusm grantRole enterprise_role=ent_connect
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.23 revokeRole

Revokes an enterprise role.

Syntax

```
revokeRole
     enterprise_role=<Enterprise role name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     [user_dn=<Distinguished name of user>]
     [group_dn=<Distinguished name of group>]
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`enterprise_role=<Enterprise role name>`	Name of the enterprise role.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`[user_dn=<Distinguished name of user>]`	DN of the user.
`[group_dn=<Distinguished name of group>]`	DN of the group.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.

Option	Description
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

Enterprise roles are access privileges assigned to enterprise users. Enterprise roles are a set of Oracle role-based authorizations across one or more databases in an enterprise domain. Enterprise roles are stored in the directory and contain one or more global roles.

A global role is a role managed in a directory, but its privileges are contained within a single database. A global role is created in a database using SQL*Plus. For example:

```
SQL> create role global_connect identified globally;
SQL> create role global_resource identified globally;
SQL> grant connect to global_connect;
SQL> grant resource to global_resource;
```

Examples

Examples include revoking an enterprise role in an enterprise domain in the realm from a user with and without SSL port connectivity to OID.

Example 6-73 Revoking a Role from a User with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm revokeRole enterprise_role=ent_connect domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-74 Revoking a Role from a User with SSL Port Conectivity to OID

```
eusm revokeRole enterprise_role=ent_connect domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-75 Revoking a Role from a User with non-SSL Port Conectivity to OID

```
eusm revokeRole enterprise_role=ent_connect domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.24 listEnterpriseRoles

Lists the enterprise roles.

Syntax

```
listEnterpriseRoles
    domain_name=<domain name>
    realm_dn=<DN of the realm>
    ldap_host=<OID host>
    ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
    keystore=<path to keystore>
    keystore_alias=<keystore password alias>
    ldap_user_dn=<DN of OID user>
    ldap_alias=<OID user password alias>
    wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Enterprise roles are access privileges assigned to enterprise users. Enterprise roles are a set of Oracle role-based authorizations across one or more databases in an enterprise domain. Enterprise roles are stored in the directory and contain one or more global roles.

A global role is a role managed in a directory, but its privileges are contained within a single database. A global role is created in a database using SQL*Plus. For example:

```
SQL> create role global_connect identified globally;
SQL> create role global_resource identified globally;
```

```
SQL> grant connect to global_connect;
SQL> grant resource to global_resource;
```

Examples

Examples include listing enterprise roles in an enterprise domain in the realm with and without SSL port connectivity to OID.

Example 6-76 List the Enterprisre Roles with SSL Port Conectivity to OID and Use Passwords Stored in the Oracle Wallet

```
eusm listEnterpriseRoles domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/oracle/
product/db_1/wallets
```

Example 6-77 List the Enterprisre Roles with SSL Port Conectivity to OID

```
eusm listEnterpriseRoles domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-78 List the Enterprisre Roles with non-SSL Port Conectivity to OID

```
eusm listEnterpriseRoles domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.25 listEnterpriseRolesOfUser

Lists the enterprise roles of a user.

Syntax

```
listEnterpriseRolesOfUser
     user_dn=<Distinguished name of user>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
user_dn=<Distinguished name of user>	DN of the user.
realm_dn=<DN of the realm>	DN of the realm.
ldap_host=<OID host>	OID host.
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>	OID non ssl port or OID ssl port.
keystore=<path to keystore>	Path to the keystore.
keystore_alias=<keystore password alias>	Keystore password taken from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

Enterprise roles are access privileges assigned to enterprise users. Enterprise roles are a set of Oracle role-based authorizations across one or more databases in an enterprise domain. Enterprise roles are stored in the directory and contain one or more global roles.

A global role is a role managed in a directory, but its privileges are contained within a single database. A global role is created in a database using SQL*Plus. For example:

```
SQL> create role global_connect identified globally;
SQL> create role global_resource identified globally;
SQL> grant connect to global_connect;
SQL> grant resource to global_resource;
```

Examples

Examples include listing the enterprise roles of a user in the realm with and without SSL port connectivity to OID.

Example 6-79 List the Enterprise Roles of a User with SSL Port Conectivity to OID and Use Passwords Stored in the Oracle Wallet

```
eusm listEnterpriseRolesOfUser
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-80 List the Enterprise Roles of a User with SSL Port Conectivity to OID

```
eusm listEnterpriseRolesOfUser
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
```

```
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_ssl_port=3131 keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-81 List the Enterprise Roles of a User with non-SSL Port Conectivity to OID

```
eusm listEnterpriseRolesOfUser
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.26 listEnterpriseRoleInfo

Lists the enterprise role information.

Syntax

```
listEnterpriseRoleInfo
     enterprise_role=<Enterprise role name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`enterprise_role=<Enterprise role name>`	Name of the enterprise role.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Enterprise roles are access privileges assigned to enterprise users. Enterprise roles are a set of Oracle role-based authorizations across one or more databases in an enterprise domain. Enterprise roles are stored in the directory and contain one or more global roles.

A global role is a role managed in a directory, but its privileges are contained within a single database. A global role is created in a database using SQL*Plus. For example:

```
SQL> create role global_connect identified globally;
SQL> create role global_resource identified globally;
SQL> grant connect to global_connect;
SQL> grant resource to global_resource;
```

Examples

Examples include listing the enterprise role information in an enterprise domain in the realm with and without SSL port connectivity to OID.

Example 6-82 List the Enterprise Role Information with SSL Port Conectivity to OID and Use Passwords Stored in the Oracle Wallet

```
eusm listEnterpriseRoleInfo enterprise_role=ent_connect
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-83 List the Enterprise Role Information with SSL Port Conectivity to OID

```
eusm listEnterpriseRoleInfo enterprise_role=ent_connect
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_ssl_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-84 List the Enterprise Role Information with non-SSL Port Conectivity to OID

```
eusm listEnterpriseRoleInfo enterprise_role=ent_connect
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.27 listGlobalRolesInDB

Lists the global roles in the database.

Syntax

```
listGlobalRolesInDB
     dbuser=<Database user name to connect>
```

```
db_alias=<Database username password alias>
dbconnect_string=<Database connect string>
wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`dbuser=<Database user name to connect>`	The database user name to connect.
`db_alias=<Database username password alias>`	Password of the database user taken from the Oracle wallet.
`dbconnect_string=<Database connect string>`	Database connect string
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Enterprise roles are access privileges assigned to enterprise users. Enterprise roles are a set of Oracle role-based authorizations across one or more databases in an enterprise domain. Enterprise roles are stored in the directory and contain one or more global roles.

A global role is a role managed in a directory, but its privileges are contained within a single database. A global role is created in a database using SQL*Plus. For example:

```
SQL> create role global_connect identified globally;
SQL> create role global_resource identified globally;
SQL> grant connect to global_connect;
SQL> grant resource to global_resource;
```

Examples

Listing the global roles for a database user.

Example 6-85 Listing the Global Roles in the Database and Using Passwords Stored in the Oracle Wallet

```
eusm listGlobalRolesInDB dbuser=system db_alias=dbadmin1
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1 wallet_location=/
oracle/product/db_1/wallets
```

Example 6-86 Listing the Global Roles in the Database

```
eusm listGlobalRolesInDB dbuser=system dbconnect_string=zzzz10-
yy.yy.company.com:1531:dbtest1
```

6.3.28 listSharedSchemasInDB

Lists the shared schemas in the database.

Syntax

```
listSharedSchemasInDB
     dbuser=<Database user name to connect>
     db_alias=<Database username password alias>
     dbconnect_string=<Database connect string>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`dbuser=<Database username to connect>`	The database user name to connect.
`db_alias=<Database username password alias>`	Password of the database user taken from the Oracle wallet.
`dbconnect_string=<Database connect string>`	Database connect string
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Listing the shared schemas for a database user.

Example 6-87 List the Shared Schemas in the Database and Use Passwords Stored in the Oracle Wallet

```
eusm listSharedSchemasInDB dbuser=system db_alias=dbadmin1
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-88 List the Shared Schemas in the Database

```
eusm listSharedSchemasInDB dbuser=system dbconnect_string=zzzz10-
yy.yy.company.com:1531:dbtest1
```

6.3.29 createProxyPerm

Creates a proxy permission object.

Syntax

```
createProxyPerm
     proxy_permission=<proxy permission name>
     domain_name=<domain name>
```

```
realm_dn=<DN of the realm>
ldap_host=<OID host>
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
keystore=<path to keystore>
keystore_alias=<keystore password alias>
ldap_user_dn=<DN of OID user>
ldap_alias=<OID user password alias>
wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`proxy_permission=<proxy permission name>`	Name of the proxy permission.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Proxy Authentication is a process typically employed in an environment with a middle tier such as a firewall, wherein the end user authenticates to the middle tier, which then authenticates to the directory on the user's behalf—as its proxy. The middle tier logs into the directory as a proxy user. A proxy user can switch identities and, once logged into the directory, switch to the end user's identity. It can perform operations on the end user's behalf, using the authorization appropriate to that particular end user.

To create the proxy permission, you must first create the proxy user in the database.

```
SQL> create user proxy_test identified by proxy_test;
SQL> alter user proxy_test grant connect through enterprise users;
```

Examples

Examples include creating a proxy permission object in an enterprise domain in the realm with and without SSL port connectivity to OID.

Example 6-89 Create the Proxy Permission Object PROXY01 with SSL Port Conectivity to OID and Use Passwords Stored in the Oracle Wallet

```
eusm createProxyPerm proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-90 Create the Proxy Permission Object PROXY01 with SSL Port Conectivity to OID

```
eusm createProxyPerm proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/
keyStore -K ldap_user_dn=cn=orcladmin
```

Example 6-91 Create the Proxy Permission Object PROXY01 with non-SSL Port Conectivity to OID

```
eusm createProxyPerm proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.30 deleteProxyPerm

Deletes a proxy permission object.

Syntax

```
deleteProxyPerm
     proxy_permission=<proxy permission name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`proxy_permission=<proxy permission name>`	Name of the proxy permission.
`domain_name=<domain name>`	Name of the domain.

Option	Description
realm_dn=<DN of the realm>	DN of the realm.
ldap_host=<OID host>	OID host.
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>	OID non ssl port or OID ssl port.
keystore=<path to keystore>	Path to the keystore.
keystore_alias=<keystore password alias>	Keystore password taken from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

Proxy Authentication is a process typically employed in an environment with a middle tier such as a firewall, wherein the end user authenticates to the middle tier, which then authenticates to the directory on the user's behalf—as its proxy. The middle tier logs into the directory as a proxy user. A proxy user can switch identities and, once logged into the directory, switch to the end user's identity. It can perform operations on the end user's behalf, using the authorization appropriate to that particular end user.

Examples

Examples include deleting a proxy permission object in an enterprise domain in the realm with and without SSL port connectivity to OID.

Example 6-92 Deleting the Proxy Permission PROXY01 with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm deleteProxyPerm proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-93 Deleting the Proxy Permission PROXY01 with SSL Port Conectivity to OID

```
eusm deleteProxyPerm proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-94 Deleting the Proxy Permission PROXY01 with non-SSL Port Conectivity to OID

```
eusm deleteProxyPerm proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.31 addTargetUser

Adds a target database user to the proxy permission object.

Syntax

```
addTargetUser
    proxy_permission=<proxy permission name>
    domain_name=<domain name>
    realm_dn=<DN of the realm>
    database_name=<Database name>
    target_user=<Target user in database>
    dbuser=<Database user name to connect>
    db_alias=<Database username password alias>
    dbconnect_string=<Database connect string>
    ldap_host=<OID host>
    ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
    keystore=<path to keystore>
    keystore_alias=<keystore password alias>
    ldap_user_dn=<DN of OID user>
    ldap_alias=<OID user password alias>
    wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`proxy_permission=<proxy permission name>`	Name of the proxy permission.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`database_name=<Database name>`	Database name.
`target_user=<Target user in database>`	Target user in the database.
`dbuser=<Database user name to connect>`	The database user name to connect.
`db_alias=<Database username password alias>`	Password of the database user taken from the Oracle wallet.
`dbconnect_string=<Database connect string>`	Database connect string
`ldap_host=<OID host>`	OID host.

Option	Description
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Proxy Authentication is a process typically employed in an environment with a middle tier such as a firewall, wherein the end user authenticates to the middle tier, which then authenticates to the directory on the user's behalf—as its proxy. The middle tier logs into the directory as a proxy user. A proxy user can switch identities and, once logged into the directory, switch to the end user's identity. It can perform operations on the end user's behalf, using the authorization appropriate to that particular end user.

Examples

Examples include adding a target database user to the proxy permission object in an enterprise domain in the realm with and without SSL port connectivity to OID.

Example 6-95 Add the Target Database User to the Proxy Permission Object with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm addTargetUser proxy_permission=PROXY01 domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
target_user=PROXY_TEST dbuser=system db_alias=dbadmin1
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-96 Add the Target Database User to the Proxy Permission Object with SSL Port Conectivity to OID

```
eusm addTargetUser proxy_permission=PROXY01 domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com database_name=dbtest1
target_user=PROXY_TEST dbuser=system dbconnect_string=zzzz10-
yy.yy.company.com:1531:dbtest1 ldap_host=xxxxx.zz.company.com ldap_port=3131
keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-97 Add the Target Database User to the Proxy Permission Object with non-SSL Port Conectivity to OID

```
eusm addTargetUser proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 target_user=PROXY_TEST dbuser=system
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.32 removeTargetUser

Removes a target database user from the proxy permission object.

Syntax

```
removeTargetUser
     proxy_permission=<proxy permission name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     database_name=<Database name>
     target_user=<Target user in database>
     dbuser=<Database user name to connect>
     db_alias=<Database username password alias>
     dbconnect_string=<Database connect string>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`proxy_permission=<proxy permission name>`	Name of the proxy permission.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`database_name=<Database name>`	Database name.
`target_user=<Target user in database>`	Target user in the database.
`dbuser=<Database username to connect>`	The database user name to connect.
`db_alias=<Database username password alias>`	Password of the database user taken from the Oracle wallet.
`dbconnect_string=<Database connect string>`	Database connect string

Option	Description
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Proxy Authentication is a process typically employed in an environment with a middle tier such as a firewall, wherein the end user authenticates to the middle tier, which then authenticates to the directory on the user's behalf—as its proxy. The middle tier logs into the directory as a proxy user. A proxy user can switch identities and, once logged into the directory, switch to the end user's identity. It can perform operations on the end user's behalf, using the authorization appropriate to that particular end user.

Examples

Examples include removing a target database user from the proxy permission object in an enterprise domain in the realm with and without SSL port connectivity to OID.

Example 6-98 Removing the Target User from the Proxy Permission Object with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm removeTargetUser proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 target_user=PROXY_TEST dbuser=system db_alias=dbadmin1
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-99 Removing the Target User from the Proxy Permission Object with SSL Port Conectivity to OID

```
eusm removeTargetUser proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 target_user=PROXY_TEST dbuser=system
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-100 Removing the Target User from the Proxy Permission Object with non-SSL Port Conectivity to OID

```
eusm removeTargetUser proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
database_name=dbtest1 target_user=PROXY_TEST dbuser=system
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.33 grantProxyPerm

Maps an enterprise user to the database user through the proxy permission object.

Syntax

```
grantProxyPerm
     proxy_permission=<proxy permission name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     [user_dn=<Distinguished name of user>]
     [group_dn=<Distinguished name of group>]
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`proxy_permission=<proxy permission name>`	Name of the proxy permission.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`[user_dn=<Distinguished name of user>]`	DN of the user.
`[group_dn=<Distinguished name of group>]`	DN of the group.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.

Option	Description
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

Proxy Authentication is a process typically employed in an environment with a middle tier such as a firewall, wherein the end user authenticates to the middle tier, which then authenticates to the directory on the user's behalf—as its proxy. The middle tier logs into the directory as a proxy user. A proxy user can switch identities and, once logged into the directory, switch to the end user's identity. It can perform operations on the end user's behalf, using the authorization appropriate to that particular end user.

Examples

Examples include mapping the enterprise user to the database user through the proxy permission object in the realm with and without SSL port connectivity to OID.

Example 6-101 Mapping the Enterprise User to the Database User Through the PROXY01 Permission Object with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm grantProxyPerm proxy_permission=PROXY01 domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-102 Mapping the Enterprise User to the Database User Through the PROXY01 Permission Object with SSL Port Conectivity to OID

```
eusm grantProxyPerm proxy_permission=PROXY01 domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-103 Mapping the Enterprise User to the Database User Through the PROXY01 Permission Object with non-SSL Port Conectivity to OID

```
eusm grantProxyPerm proxy_permission=PROXY01 domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.34 revokeProxyPerm

Revokes a proxy permission object.

Syntax

```
revokeProxyPerm
     proxy_permission=<proxy permission name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     [user_dn=<Distinguished name of user>]
     [group_dn=<Distinguished name of group>]
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`proxy_permission=<proxy permission name>`	Name of the proxy permission.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`[user_dn=<Distinguished name of user>]`	DN of the user.
`[group_dn=<Distinguished name of group>]`	DN of the group.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Proxy Authentication is a process typically employed in an environment with a middle tier such as a firewall, wherein the end user authenticates to the middle tier, which then authenticates to the directory on the user's behalf—as its proxy. The middle tier logs into the directory as a proxy user. A proxy user can switch identities and, once logged into the directory, switch to the end user's identity. It can perform operations on the end user's behalf, using the authorization appropriate to that particular end user.

Examples

Examples include revoking proxy permission object PROXY01 from the database user in the realm with and without SSL port connectivity to OID.

Example 6-104 Revoking Proxy Permission Object PROXY01 From the User with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm revokeProxyPerm proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-105 Revoking Proxy Permission Object PROXY01 From the User with SSL Port Conectivity to OID

```
eusm revokeProxyPerm proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-106 Revoking Proxy Permission Object PROXY01 From the User with non-SSL Port Conectivity to OID

```
eusm revokeProxyPerm proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.35 listProxyPermissions

Lists proxy permissions. Input is the domain name.

Syntax

```
listProxyPermissions
    domain_name=<domain name>
    realm_dn=<DN of the realm>
    ldap_host=<OID host>
    ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
```

```
keystore=<path to keystore>
keystore_alias=<keystore password alias>
ldap_user_dn=<DN of OID user>
ldap_alias=<OID user password alias>
wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Proxy Authentication is a process typically employed in an environment with a middle tier such as a firewall, wherein the end user authenticates to the middle tier, which then authenticates to the directory on the user's behalf—as its proxy. The middle tier logs into the directory as a proxy user. A proxy user can switch identities and, once logged into the directory, switch to the end user's identity. It can perform operations on the end user's behalf, using the authorization appropriate to that particular end user.

Examples

Examples include listing the proxy permissions for the enterprise domain in the realm with and without SSL port connectivity to OID.

Example 6-107 Listing the Proxy Permissions for the Domain with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listProxyPermissions domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/
oracle/product/db_1/wallets
```

Example 6-108 Listing the Proxy Permissions for the Domain with SSL Port Conectivity to OID

```
eusm listProxyPermissions domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-109 Listing the Proxy Permissions for the Domain with non-SSL Port Conectivity to OID

```
eusm listProxyPermissions domain_name=OracleDefaultDomain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.36 listProxyPermissionsOfUser

Lists the proxy permissions for the user. Input is the user distinguished name.

Syntax

```
listProxyPermissionsOfUser
     user_dn=<Distinguished name of user>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`user_dn=<Distinguished name of user>`	DN of the user.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Proxy Authentication is a process typically employed in an environment with a middle tier such as a firewall, wherein the end user authenticates to the middle tier, which then authenticates to the directory on the user's behalf—as its proxy. The middle tier logs into the directory as a proxy user. A proxy user can switch identities and, once logged into the directory, switch to the end user's identity. It can perform operations on the end user's behalf, using the authorization appropriate to that particular end user.

Examples

Examples include listing the proxy permissions for the user in the realm with and without SSL port connectivity to OID.

Example 6-110 List the Proxy Permission for the User with SSL Port Conectivity to OID and Use Passwords Stored in the Oracle Wallet

```
eusm listProxyPermissionsOfUser
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/
oracle/product/db_1/wallets
```

Example 6-111 List the Proxy Permission for the User with SSL Port Conectivity to OID

```
eusm listProxyPermissionsOfUser
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-112 List the Proxy Permission for the User with non-SSL Port Conectivity to OID

```
eusm listProxyPermissionsOfUser
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.37 listProxyPermissionInfo

Lists the proxy permission information.

Syntax

```
listProxyPermissionInfo
     proxy_permission=<proxy permission name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
```

```
keystore=<path to keystore>
keystore_alias=<keystore password alias>
ldap_user_dn=<DN of OID user>
ldap_alias=<OID user password alias>
wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`proxy_permission=<proxy permission name>`	Name of the proxy permission.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

Proxy Authentication is a process typically employed in an environment with a middle tier such as a firewall, wherein the end user authenticates to the middle tier, which then authenticates to the directory on the user's behalf—as its proxy. The middle tier logs into the directory as a proxy user. A proxy user can switch identities and, once logged into the directory, switch to the end user's identity. It can perform operations on the end user's behalf, using the authorization appropriate to that particular end user.

Examples

Examples include listing the proxy permission information for the enterprise domain in the realm with and without SSL port connectivity to OID.

Example 6-113 List Proxy Permission Information with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listProxyPermissionInfo proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-114 List Proxy Permission Information with SSL Port Conectivity to OID

```
eusm listProxyPermissionInfo proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-115 List Proxy Permission Information with non-SSL Port Conectivity to OID

```
eusm listProxyPermissionInfo proxy_permission=PROXY01
domain_name=OracleDefaultDomain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.38 listTargetUsersInDB

Lists the target users in the database.

Syntax

```
listTargetUsersInDB
     dbuser=<Database user name to connect>
     db_alias=<Database username password alias>
     dbconnect_string=<Database connect string>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`dbuser=<Database user name to connect>`	The database user name to connect.
`db_alias=<Database username password alias>`	Password of the database user taken from the Oracle wallet.
`dbconnect_string=<Database connect string>`	Database connect string
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Listing the target users in the database.

Example 6-116 Listing the Target Users in the Database and Using Passwords Stored in the Oracle Wallet

```
eusm listTargetUsersInDB dbuser=system db_alias=dbadmin1
dbconnect_string=zzzz10-yy.yy.company.com:1531:dbtest1 wallet_location=/
oracle/product/db_1/wallets
```

Example 6-117 Listing the Target Users in the Database

```
eusm listTargetUsersInDB dbuser=system dbconnect_string=zzzz10-
yy.yy.company.com:1531:dbtest1
```

6.3.39 setDBOIDAuth

Sets the database-OID authentication method.

Syntax

```
setDBOIDAuth
     realm_dn=<DN of the realm>
     dboid_auth=<Default DB OID authentication>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`realm_dn=<DN of the realm>`	DN of the realm.
`dboid_auth=<Default DB OID authentication>`	Default DB OID authentication.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

The OID authentication method can be either SSL or PASSWORD.

Examples

Examples include setting the database-OID authentication method in the realm with and without SSL port connectivity to OID.

Example 6-118 Setting the Database-OID Authentication Method with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm setDBOIDAuth realm_dn=dc=yy,dc=company,dc=com dboid_auth=SSL
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-119 Setting the Database-OID Authentication Method with SSL Port Conectivity to OID

```
eusm setDBOIDAuth realm_dn=dc=yy,dc=company,dc=com dboid_auth=SSL
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-120 Setting the Database-OID Authentication Method with non-SSL Port Conectivity to OID

```
eusm setDBOIDAuth realm_dn=dc=yy,dc=company,dc=com dboid_auth=SSL
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.40 listDBOIDAuth

Lists the database-OID authentication method.

Syntax

```
listDBOIDAuth
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
realm_dn=<DN of the realm>	DN of the realm.
ldap_host=<OID host>	OID host.
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>	OID non ssl port or OID ssl port.
keystore=<path to keystore>	Path to the keystore.
keystore_alias=<keystore password alias>	Keystore password taken from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

The OID authentication method can be either SSL or PASSWORD.

Examples

Examples include listing the database-OID authentication method in the realm with and without SSL port connectivity to OID.

Example 6-121 Listing the Database-OID Authentication Method with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listDBOIDAuth realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-122 Listing the Database-OID Authentication Method with SSL Port Conectivity to OID

```
eusm listDBOIDAuth realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-123 Listing the Database-OID Authentication Method with non-SSL Port Conectivity to OID

```
eusm listDBOIDAuth realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.41 addToPwdAccessibleDomains

Adds a domain to the password accessible domains group in the realm.

Syntax

```
addToPwdAccessibleDomains
     realm_dn=<DN of the realm>
     domain_name=<name of enterprise domain>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`realm_dn=<DN of the realm>`	DN of the realm.
`domain_name=<domain name>`	Name of the domain.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include adding to password accessible domains in the realm with and without SSL port connectivity to OID.

Example 6-124 Adding to Password Accessible Domains with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm addToPwdAccessibleDomains domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/
oracle/product/db_1/wallets
```

Example 6-125 Adding to Password Accessible Domains with SSL Port Conectivity to OID

```
eusm addToPwdAccessibleDomains domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-126 Adding to Password Accessible Domains with non-SSL Port Conectivity to OID

```
eusm addToPwdAccessibleDomains domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.42 removeFromPwdAccessibleDomains

Removes a domain from the password accessible domains group in the realm.

Syntax

```
removeFromPwdAccessibleDomains
     realm_dn=<DN of the realm>
     domain_name=<name of enterprise domain>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space

Option	Description
`realm_dn=<DN of the realm>`	DN of the realm.
`domain_name=<domain name>`	Name of the domain.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include removing from password accessible domains in the realm with and without SSL port connectivity to OID.

Example 6-127 Removing from Password Accessible Domains with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm removeFromPwdAccessibleDomains domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/
oracle/product/db_1/wallets
```

Example 6-128 Removing from Password Accessible Domains with SSL Port Conectivity to OID

```
eusm removeFromPwdAccessibleDomains domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-129 Removing from Password Accessible Domains with non-SSL Port Conectivity to OID

```
eusm removeFromPwdAccessibleDomains domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.43 listPwdAccessibleDomains

Lists the password accessible domains in the realm.

Syntax

```
listPwdAccessibleDomains
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
realm_dn=<DN of the realm>	DN of the realm.
ldap_host=<OID host>	OID host.
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>	OID non ssl port or OID ssl port.
keystore=<path to keystore>	Path to the keystore.
keystore_alias=<keystore password alias>	Keystore password taken from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include listing the password accessible domains in the realm with and without SSL port connectivity to OID.

Example 6-130 Listing the Password Accessible Domains with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listPwdAccessibleDomains realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-131 Listing the Password Accessible Domains with SSL Port Conectivity to OID

```
eusm listPwdAccessibleDomains realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-132 Listing the Password Accessible Domains with non-SSL Port Conectivity to OID

```
eusm listPwdAccessibleDomains realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.44 listRealmCommonAttr

Lists the realm common attributes.

Syntax

```
listRealmCommonAttr
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include listing the realm common attributes with and without SSL port connectivity to OID.

Example 6-133 Listing the Realm Common Attributes with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listRealmCommonAttr realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-134 Listing the Realm Common Attributes with SSL Port Conectivity to OID

```
eusm listRealmCommonAttr realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-135 Listing the Realm Common Attributes with non-SSL Port Conectivity to OID

```
eusm listRealmCommonAttr realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.45 createAppCtxNamespace

Adds a new namespace.

Syntax

```
createAppCtxNamespace
     namespace=<namespace name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`namespace=<namespace name>`	Name of the namespace.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken interactively at the prompt or from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken interactively at the prompt or from the Oracle wallet.

Option	Description
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include adding a new domain namespace in the realm with and without SSL port connectivity to OID.

Example 6-136 Adding a New Domain Namespace with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm createAppCtxNamespace namespace=ns1 domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/
oracle/product/db_1/wallets
```

Example 6-137 Adding a New Domain Namespace with SSL Port Conectivity to OID

```
eusm createAppCtxNamespace namespace=ns1 domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-138 Adding a New Domain Namespace with non-SSL Port Conectivity to OID

```
eusm createAppCtxNamespace namespace=ns1 domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.46 deleteAppCtxNamespace

Deletes a namespace.

Syntax

```
deleteAppCtxNamespace
     namespace=<namespace name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
```

```
ldap_alias=<OID user password alias>
wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`namespace=<namespace name>`	Name of the namespace.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include deleting a domain namespace from the realm with and without SSL port connectivity to OID.

Example 6-139 Deleting a Domain Namespace with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm deleteAppCtxNamespace namespace=ns1 domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/oracle/
product/db_1/wallets
```

Example 6-140 Deleting a Domain Namespace with SSL Port Conectivity to OID

```
eusm deleteAppCtxNamespace namespace=ns1 domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-141 Deleting a Domain Namespace with non-SSL Port Conectivity to OID

```
eusm deleteAppCtxNamespace namespace=ns1 domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.47 listAppCtxNamespaces

Lists the namespaces.

Syntax

```
listAppCtxNamespaces
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include listing the domain namespaces in the realm with and without SSL port connectivity to OID.

Example 6-142 Listing the Namespaces with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listAppCtxNamespaces domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/oracle/
product/db_1/wallets
```

Example 6-143 Listing the Namespaces with SSL Port Conectivity to OID

```
eusm listAppCtxNamespaces domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-144 Listing the Namespaces with non-SSL Port Conectivity to OID

```
eusm listAppCtxNamespaces domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.48 createAppCtxAttribute

Adds a new attribute.

Syntax

```
createAppCtxAttribute
     attribute_name=<attribute name>
     namespace=<namespace name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
attribute_name=<attribute name>	Name of the attribute.
namespace=<namespace name>	Name of the namespace.
domain_name=<domain name>	Name of the domain.
realm_dn=<DN of the realm>	DN of the realm.
ldap_host=<OID host>	OID host.
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>	OID non ssl port or OID ssl port.
keystore=<path to keystore>	Path to the keystore.
keystore_alias=<keystore password alias>	Keystore password taken from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include adding a new attribute to a domain namespace in the realm with and without SSL port connectivity to OID.

Example 6-145 Adding a New Attribute with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm createAppCtxAttribute attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-146 Adding a New Attribute with SSL Port Conectivity to OID

```
eusm createAppCtxAttribute attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-147 Adding a New Attribute with non-SSL Port Conectivity to OID

```
eusm createAppcCtxAttribute attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.49 deleteAppCtxAttribute

Deletes an attribute.

Syntax

```
deleteAppCtxAttribute
     attribute_name=<attribute name>
     namespace=<namespace name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`attribute_name=<attribute name>`	Name of the attribute.
`namespace=<namespace name>`	Name of the namespace.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include deleting an attribute from a domain namespace in the realm with and without SSL port connectivity to OID.

Example 6-148 Deleting Attributes with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm deleteAppCtxAttribute attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-149 Deleting Attributes with SSL Port Conectivity to OID

```
eusm deleteAppCtxAttribute attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-150 Deleting Attributes with non-SSL Port Conectivity to OID

```
eusm deleteAppCtxAttribute attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.50 listAppCtxAttributes

Lists the attributes.

Syntax

```
listAppCtxAttributes
     namespace=<namespace name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`namespace=<namespace name>`	Name of the namespace.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.

Option	Description
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>	OID non ssl port or OID ssl port.
keystore=<path to keystore>	Path to the keystore.
keystore_alias=<keystore password alias>	Keystore password taken from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include listing the domain namespace attributes in the realm with and without SSL port connectivity to OID.

Example 6-151 Listing Attributes with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listAppCtxAttributes namespace=ns1 domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/oracle/
product/db_1/wallets
```

Example 6-152 Listing Attributes with SSL Port Conectivity to OID

```
eusm listAppCtxAttributes namespace=ns1 domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-153 Example Title with non-SSL Port Conectivity to OID

```
eusm listAppCtxAttributes namespace=ns1 domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.51 createAppCtxAttributeValue

Adds a new attribute value.

Syntax

```
createAppCtxAttributeValue
     attribute_value=<value of the attribute>
     attribute_name=<attribute name>
     namespace=<namespace name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`attribute_value=<value of the attribute>`	Value of the attribute.
`attribute_name=<attribute name>`	Name of the attribute.
`namespace=<namespace name>`	Name of the namespace.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include adding a new attribute value to an attribute to a domain namespace in the realm with and without SSL port connectivity to OID.

Example 6-154 Adding a New Attribute Value with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm createAppCtxAttributeValue attribute_value=val1 attribute_name=attr1
namespace=ns1 domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-155 Adding a New Attribute Value with SSL Port Conectivity to OID

```
eusm createAppCtxAttributeValue attribute_value=val1 attribute_name=attr1
namespace=ns1 domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-156 Adding a New Attribute Value with non-SSL Port Conectivity to OID

```
eusm createAppCtxAttributeValue attribute_value=val1 attribute_name=attr1
namespace=ns1 domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.52 deleteAppCtxAttributeValue

Deletes an attribute value.

Syntax

```
deleteAppCtxAttributeValue
     attribute_value=<value of the attribute>
     attribute_name=<attribute name>
     namespace=<namespace name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`attribute_value=<value of the attribute>`	Value of the attribute.
`attribute_name=<attribute name>`	Name of the attribute.

Option	Description
namespace=<namespace name>	Name of the namespace.
domain_name=<domain name>	Name of the domain.
realm_dn=<DN of the realm>	DN of the realm.
ldap_host=<OID host>	OID host.
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>	OID non ssl port or OID ssl port.
keystore=<path to keystore>	Path to the keystore.
keystore_alias=<keystore password alias>	Keystore password taken from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include deleting an attribute value from an attribute in a domain namespace in the realm with and without SSL port connectivity to OID.

Example 6-157 Deleting an Attribute Value with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm deleteAppCtxAttributeValue attribute_value=val1
attribute_name=attr1 namespace=ns1 domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore keystore_alias=keystore1
ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1 wallet_location=/
oracle/product/db_1/wallets
```

Example 6-158 Deleting an Attribute Value with SSL Port Conectivity to OID

```
eusm deleteAppCtxAttributeValue attribute_value=val1
attribute_name=attr1 namespace=ns1 domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3131 keystore=/etc/myapp/keyStore ldap_user_dn=cn=orcladmin
```

Example 6-159 Deleting an Attribute Value with non-SSL Port Conectivity to OID

```
eusm deleteAppCtxAttributeValue attribute_value=val1
attribute_name=attr1 namespace=ns1 domain_name=test_domain
realm_dn=dc=yy,dc=company,dc=com ldap_host=xxxxx.zz.company.com
ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.53 listAppCtxAttributeValues

Lists the attribute values.

Syntax

```
listAppCtxAttributeValues
     attribute_name=<attribute name>
     namespace=<namespace name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`attribute_name=<attribute name>`	Name of the attribute.
`namespace=<namespace name>`	Name of the namespace.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include listing the attribute values for an attribute for a domain namespace in the realm with and without SSL port connectivity to OID.

Example 6-160 Listing the Attribute Values with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listAppCtxAttributeValues attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/
keyStore keystore_alias=keystore1 ldap_user_dn=cn=orcladmin
ldap_alias=ldabadmin1 wallet_location=/oracle/product/db_1/wallets
```

Example 6-161 Listing the Attribute Values with SSL Port Conectivity to OID

```
eusm listAppCtxAttributeValues attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/
keyStore ldap_user_dn=cn=orcladmin
```

Example 6-162 Listing the Attribute Values with non-SSL Port Conectivity to OID

```
eusm listAppCtxAttributeValues attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.54 createAppCtxUsers

Adds a new user for an attribute value.

Syntax

```
createAppCtxUsers
     user_dn=<Distinguished name of user>
     attribute_value=<value of the attribute>
     attribute_name=<attribute name>
     namespace=<namespace name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`user_dn=<Distinguished name of user>`	DN of the user.

Option	Description
attribute_value=<value of the attribute>	Value of the attribute.
attribute_name=<attribute name>	Name of the attribute.
namespace=<namespace name>	Name of the namespace.
domain_name=<domain name>	Name of the domain.
realm_dn=<DN of the realm>	DN of the realm.
ldap_host=<OID host>	OID host.
ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>	OID non ssl port or OID ssl port.
keystore=<path to keystore>	Path to the keystore.
keystore_alias=<keystore password alias>	Keystore password taken from the Oracle wallet.
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include adding a new user for an attribute value to an attribute to a domain namespace in the realm with and without SSL port connectivity to OID.

Example 6-163 Adding a New User for an Attribute Value with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm createAppCtxUsers user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
attribute_value=val1 attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-164 Adding a New User for an Attribute Value with SSL Port Conectivity to OID

```
eusm createAppCtxUsers user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
attribute_value=val1 attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-165 Adding a New User for an Attribute Value with non-SSL Port Conectivity to OID

```
eusm createAppCtxUsers
user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
attribute_value=val1 attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.55 deleteAppCtxUsers

Deletes a user from an attribute value.

Syntax

```
deleteAppCtxUsers
     user_dn=<Distinguished name of user>
     attribute_value=<value of the attribute>
     attribute_name=<attribute name>
     namespace=<namespace name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`user_dn=<Distinguished name of user>`	DN of the user.
`attribute_value=<value of the attribute>`	Value of the attribute.
`attribute_name=<attribute name>`	Name of the attribute.
`namespace=<namespace name>`	Name of the namespace.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.

Option	Description
ldap_user_dn=<DN of OID user>	DN of OID user which is used for authenticating and executing a command in the OID.
ldap_alias=<OID user password alias>	OID user password taken from the Oracle wallet.
wallet_location=<wallet location>	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include deleting a user from an attribute value for an attribute in a domain namespace in the realm with and without SSL port connectivity to OID.

Example 6-166 Deleting a User from an Attribute Value with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm deleteAppCtxUsers user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
attribute_value=val1 attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-167 Deleting a User from an Attribute Value with SSL Port Conectivity to OID

```
eusm deleteAppCtxUsers user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
attribute_value=val1 attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-168 Deleting a User from an Attribute Value with non-SSL Port Conectivity to OID

```
eusm deleteAppCtxUsers user_dn=cn=test_user,cn=Users,dc=yy,dc=company,dc=com
attribute_value=val1 attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

6.3.56 listAppCtxUsers

Lists all users for an attribute value.

Syntax

```
listAppCtxUsers
     attribute_value=<value of the attribute>
     attribute_name=<attribute name>
     namespace=<namespace name>
     domain_name=<domain name>
     realm_dn=<DN of the realm>
     ldap_host=<OID host>
     ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>
     keystore=<path to keystore>
     keystore_alias=<keystore password alias>
     ldap_user_dn=<DN of OID user>
     ldap_alias=<OID user password alias>
     wallet_location=<wallet location>
```

Options

Each option must be prefixed with a space. Multiple options can be concatenated and prefixed with single space.

Option	Description
`attribute_value=<value of the attribute>`	Value of the attribute.
`attribute_name=<attribute name>`	Name of the attribute.
`namespace=<namespace name>`	Name of the namespace.
`domain_name=<domain name>`	Name of the domain.
`realm_dn=<DN of the realm>`	DN of the realm.
`ldap_host=<OID host>`	OID host.
`ldap_port=<OID non ssl port> | ldap_ssl_port=<OID ssl port>`	OID non ssl port or OID ssl port.
`keystore=<path to keystore>`	Path to the keystore.
`keystore_alias=<keystore password alias>`	Keystore password taken from the Oracle wallet.
`ldap_user_dn=<DN of OID user>`	DN of OID user which is used for authenticating and executing a command in the OID.
`ldap_alias=<OID user password alias>`	OID user password taken from the Oracle wallet.
`wallet_location=<wallet location>`	Path to Oracle wallet when using the wallet.

Usage Notes

None.

Examples

Examples include listing all users for an attribute value for an attribute for a domain namespace in the realm with and without SSL port connectivity to OID.

Example 6-169 Listing All Users for an Attribute Value with SSL Port Conectivity to OID and Using Passwords Stored in the Oracle Wallet

```
eusm listAppCtxUsers attribute_value=val1 attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
keystore_alias=keystore1 ldap_user_dn=cn=orcladmin ldap_alias=ldabadmin1
wallet_location=/oracle/product/db_1/wallets
```

Example 6-170 Listing All Users for an Attribute Value with SSL Port Conectivity to OID

```
eusm listAppCtxUsers attribute_value=val1 attribute_name=attr1 namespace=ns1
domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3131 keystore=/etc/myapp/keyStore
ldap_user_dn=cn=orcladmin
```

Example 6-171 Listing All Users for an Attribute Value with non-SSL Port Conectivity to OID

```
eusm listAppcCtxUsers attribute_value=val1 attribute_name=attr1
namespace=ns1 domain_name=test_domain realm_dn=dc=yy,dc=company,dc=com
ldap_host=xxxxx.zz.company.com ldap_port=3060 ldap_user_dn=cn=orcladmin
```

A

SSL External Users Conversion Script

You should run the SSL external users conversion script after upgrading to Oracle Database 12*c* Release 1 (12.1) and later, in case you were using SSL-authenticated external users in a pre-Oracle Database 10*g* Release 2 (10.2) release. The script converts SSL-authenticated external users in pre-Oracle Database 10*g* Release 2 (10.2) releases into SSL-authenticated external users in Oracle Database 12*c* Release 1 (12.1) and later.

> **Note:**
>
> The SSL external users conversion script needs to be run only if you have upgraded from a pre-Oracle Database 10*g* Release 2 (10.2) release.

About Using a Secure External Password Store

If you want to use a secure external password store, then configure the Oracle wallet as described in the information that follows; otherwise, passwords can be provided interactively and you can skip this section.

Before you run the `extusrupgrade` script, configure a client-side Oracle wallet as a secure external password store so that your applications can use password credentials stored in the wallet to connect to databases. Storing database password credentials in a client-side Oracle wallet eliminates the need to embed passwords in application code, batch jobs, or scripts. This reduces the risk of exposing passwords in the clear in scripts and application code, and allows you to more easily manage password policies for user accounts without changing application code or scripts whenever passwords change.

See Configuring a Client to Use the External Password Store for steps to configure a client to use the external password store by using the `mkstore` command-line utility.

> **Note:**
>
> The external password store of the wallet is separate from the area where public key infrastructure (PKI) credentials are stored. Use the command-line utility `mkstore` to manage these credentials.

Using the `mkstore CreateCredential` command, configure the following `dbuser` credential by providing information for `<alias, username, password>`, in which you will be prompted to enter the password for the user:

- `dbalias`, `dbuser`, *`password`*

Configuring this user credential allows you to use the following parameter on the `extusrupgrade` script command line:

- `-dbalias=<db-password-alias>`

Conversion script examples use the following wallet credential information for user `dbuser` that was provided for the alias name, user name, and password. The wallet location is specified as shown.

- `dbmanager1`, `system`, *`password`*
- `wallet_location=/oracle/product/db_1/wallets`

Conversion script examples use the following entries on the command-line:

- `-dbalias=dbmanager1`
- `wallet_location=/oracle/product/db_1/wallets`

After configuring the client-side wallet, enable auto-login for Oracle Wallets to allow the administrator running the `extusrupgrade` script to access and perform `extusrupgrade` services without having to supply the necessary credentials.

> **See Also:**
>
> - Managing the Secure External Password Store for Password Credentials for more information about creating a client-side password store wallet to store alias, user name, and password credentials for users

This chapter covers the following topics:

A.1 Using the SSL External Users Conversion Script

The SSL external users conversion script has the following syntax:

```
$ORACLE_HOME/rdbms/bin/extusrupgrade
--dbconnectstring database connect string
--dbuser database user
[-dbalias database user password alias]
[-wallet_location wallet location]
[-a]
[-l username1,username2,...]
[-f filename]
[-o]
[-h]
note:          -a upgrade all qualified users
               -l upgrade list of users seperated by comma
               -f upgrade list of users specified by the file. One user name per
line
               -o output all qualified users to standard out. Not combine with
other options
               -h show this help.
```

The *`database connect string`* should be in the format *`hostname`*:*`port_no`*:*`sid`*, where *`hostname`* is the name of the host on which the database is running, *`port_no`* is the listener port number and *`sid`* is the system identifier for the database instance.

If you have created a secure external password store using the `mkstore` command-line utility, then create the `dbuser` credential in the wallet using the `mkstore CreateCredential` command using the syntax `<alias, username, password>`. For example, `dbmanager1, system, password`.

Next, enable auto login for Oracle wallets. This allows the administrator user running the `extusrupgrade` script access to`extusrupgrade` services without having to supply the necessary credentials.

Now you can use the database alias parameter `-dbalias <database user password alias>` and the wallet location parameter `-wallet_location <wallet location>` on the command line for running the `extusrupgrade` conversion script.

The following examples assume that the wallet has a `dbuser` credential defined using the syntax `<alias, username, password>` as `dbmanager1, system, password`. For examples, the wallet location is shown as `/oracle/product/19.1.0/db_1/wallets`.

> **Note:**
>
> For information about the maximum length allowed for an encryption password, refer to the Oracle Database Utilities Guide.

Use the `-a` option to convert all SSL-authenticated external users. Here is an example:

```
extusrupgrade --dbconnectstring mymachine:1521:ORA001 --dbuser system -dbialis
dbmanager1 -wallet_location /oracle/product/db_1/wallets -a
```

Use the `-l` option to specify a comma-delimited list of users to be converted. For example:

```
extusrupgrade --dbconnectstring mymachine:1521:ORA001 --dbuser system -dbialis
dbmanager1 -wallet_location /oracle/product/db_1/wallets -l user1,user2,user3
```

Use the `-f` option to specify a file that has the list of users to be converted. For example:

```
extusrupgrade --dbconnectstring mymachine:1521:ORA001 --dbuser system -dbialis
dbmanager1 -wallet_location /oracle/product/db_1/wallets -f usernames.txt
```

There should be one user name in each line in the specified file. Here is a sample `usernames.txt` file:

```
user#1
user>2
user,3
user4
user5
```

You must use the `-f` option to convert users who have special characters (such as #) in their user names.

> **Note:**
>
> You can combine the `-l` and `-f` options in the same command. The script combines the list of users from both the `-l` and `-f` options. If you use the `-a` option along with the `-l option` and the `-f` option, then the `-a` option is ignored.

You can use the `-o` option to print a list of SSL-authenticated external users to the standard output device. The output lists the users you can convert using the *extusrupgrade* script. The `-o` option cannot be combined with any other option.

```
extusrupgrade --dbconnectstring mymachine:1521:ORA001 --dbuser system -dbialis
dbmanager1 -wallet_location /oracle/product/db_1/wallets -o
```

A sample output for this could be:

```
user1
user2
user3
```

> **Tip:**
>
> You can redirect the command output to a file to get a list of users who can be converted. You can then edit the file and use it with the `-f` option.

A.2 Converting Global Users into External Users

Oracle Database 10*g* and later allows SSL-authenticated external users and SSL-authenticated global users to coexist in the database. Previous releases had the restriction that all SSL users must be either global users or external users, depending on whether Oracle Internet Directory is being used or not for authenticating the users.

If you want a user to be able to connect to the database even when Oracle Internet Directory is not available, then the user should be configured as an external user. You can convert SSL-authenticated global users into SSL-authenticated external users by using the SSL external users conversion script.

If you have created a secure external password store using the `mkstore` command-line utility and have created the `dbuser` credential in the wallet using the `mkstore CreateCredential` command using the syntax `<alias, username, password>`. For example, `dbmanager1, system, password`. For examples, the wallet location is shown as `/oracle/product/db_1/wallets`. Now you can use the database alias parameter `-dbalias <database user password alias>` and the wallet location parameter `-wallet_location <wallet location>` on the command line when running the `extusrupgrade` conversion script. Note that if you have enabled auto login for Oracle wallets, then the administrator user running the `extusrupgrade` script can access extusrupgrade services without having to supply the necessary credentials.

For example:

```
extusrupgrade --dbconnectstring mymachine:1521:ORA001 --dbuser system -dbialis
dbmanager1 -wallet_location /oracle/product/db_1/wallets -l user1,user2
```

The preceding example converts two global users into external users.

B

Integrating Enterprise User Security with Microsoft Active Directory

This appendix lists the steps involved in integrating Enterprise User Security with Microsoft Active Directory using kerberos for authentication. It includes the following sections:

B.1 About Direct Integration with Microsoft Active Directory

Oracle Database release 18c, version 18.1 and later supports direct integration with Microsoft Active Directory (MSAD) using the new centrally managed users capability.

Beginning with Oracle Database release 18c, version 18.1, Oracle Database introduces centrally managed users to authenticate and authorize users directly with Microsoft Active Directory. With centrally managed users, users accessing the database can be centrally managed to improve an organization's security posture. An enterprise user (a user in Microsoft Active Directory) can be exclusively mapped to a database account, or many enterprise users (in an Microsoft Active Directory group) can be mapped to a shared account in the database. Microsoft Active Directory groups can also be mapped to a database global role, which provides users with additional privileges and roles above what their login account (exclusive or shared) is granted. With centrally managed users, users can be authenticated with passwords, and Kerberos and PKI certificates.

For organizations beginning new directory services projects that do not require some of the more complex Enterprise User Security features like trusted database links, centrally managed users with Microsoft Active Directory allows you to use their current Microsoft Active Directory service as their centralized user management and centralized database access authorization. For new directory services projects, this may be the preferred option in implementing directory services as this reduces complexity as well as costs of operation in terms of maintenance and development.

Enterprise users can also make use of Oracle Internet Directory, which is a part of the Oracle Identity Management infrastructure. If your organization uses a third party directory like Microsoft Active Directory to store and manage user entries, then you can integrate it with Oracle Internet Directory to manage Enterprise User Security. This may be your preferred option if your organization requires some of the more complex Enterprise User Security features like trusted database links.

> **See Also:**
>
> *Oracle Database Security Guide* for details about configuring centrally managed users with Microsoft Active Directory.

B.2 Set Up Synchronization Between Active Directory and Oracle Internet Directory

Oracle components make use of Oracle Internet Directory for centralized security administration. Your organization might have a Microsoft Windows domain that uses Microsoft Active Directory for centralized administration. You should set up synchronization between Oracle Internet Directory and Microsoft Active Directory before you configure Enterprise User Security to work with Microsoft Active Directory.

Synchronization profiles are used to synchronize the two directories. The profile contains configuration information required to synchronize the two directories. This includes direction of synchronization, mapping rules and formats, connection details of Microsoft Windows domain and the like. Mapping rules contain domain rules and attribute rules to map a domain and attributes in one directory to the other directory, optionally formatting the attributes.

> **See Also:**
>
> For step-by-step instructions on integrating Oracle Internet Directory with Microsoft Active Directory, refer to the *Oracle Identity Management Integration Guide*

B.3 Set Up Active Directory to Interoperate with Oracle Client

The following tasks must be performed on the Windows domain controller:

1. Create the Oracle Database Principal in Microsoft Active Directory.

   This creates a new user for the database in Microsoft Active Directory.

2. Use the `okcreate` command-line utility to automate the creation of the service principal `keytab` file.

   Beginning with Oracle Database 12*c* Release 2 (12.2), the `okcreate` utility provides for automation of service principal keytab creation on the Key Distribution Center (KDC) to create all service keytabs that setup requires. To use an Active Directory as a KDC, this requires the `keytab` file. If the Oracle client does not have a `keytab` file in the location specified by `SQLNET.KERBEROS5_CONF`, it uses a generic `krb5.conf` file. In this case, it detects realm and KDC settings from DNS. This utility takes input about the keytabs to create and output them to a specified location. The inputs taken are the service name (defaults to oracle) and a list of hostnames on which the database server is installed.

> **See Also:**
>
> *Oracle Database Security Guide* for a detailed listing of the preceding steps.

B.4 Set Up Oracle Database to Interoperate with Microsoft Active Directory

The following task must be performed on the host computer where Oracle Database is installed:

- Update the `sqlnet.ora` file in the database with kerberos parameters

> **See Also:**
>
> *Oracle Database Security Guide* for a detailed description of the preceding step.

B.5 Set Up Oracle Database Client to Interoperate with Microsoft Active Directory

The following steps must be performed on the Oracle kerberos client:

1. Create client kerberos configuration files

   The client kerberos configuration files refer to the Microsoft Active Directory as the kerberos KDC.

2. Specify kerberos parameters in the client `sqlnet.ora` file

   You can either manually update the file or use Oracle Net Manager utility.

> **See Also:**
>
> *Oracle Database Security Guide* for a detailed listing of the preceding steps.

B.6 Obtain an Initial Ticket for the Client

Before a client can connect to the database, the client must request for an initial ticket. The initial ticket identifies the client as having the rights to ask for additional service tickets. An initial ticket is requested using the `okinit` command.

> **See Also:**
>
> *Oracle Database Security Guide* for more details on requesting an initial ticket with `okinit`.

B.7 Configure Enterprise User Security for Kerberos Authentication

To configure Enterprise User Security for Kerberos Authentication, use the following steps:

1. Register the database in Oracle Internet Directory

   You can use Database Configuration Assistant for registering the database.

2. Configure Enterprise User Security Objects in the database and Oracle Internet Directory

   Create global schemas and global roles in the database. Also create enterprise roles in the enterprise domain. Configure user schema mappings for the enterprise domain, add global database roles to enterprise roles and grant enterprise roles to enterprise users for database access.

3. Configure the enterprise domain to accept kerberos authentication

   Use Oracle Enterprise Manager to enable kerberos authentication for your enterprise domain.

4. Connect as kerberos authenticated enterprise user.

   Launch SQL*Plus and use the command, `connect /@net_service_name` to connect as a kerberos authenticated enterprise user.

> **See Also:**
>
> For detailed information on the preceding steps, refer to "Configuring Enterprise User Security for Kerberos Authentication" .

Glossary

access control

The ability of a system to grant or limit access to specific data for specific clients or groups of clients.

Access Control Lists (ACLs)

The group of access directives that you define. The directives grant levels of access to specific data for specific clients, or groups of clients, or both.

Advanced Encryption Standard

Advanced Encryption Standard (AES) is a new cryptographic algorithm that has been approved by the National Institute of Standards and Technology as a replacement for DES. The AES standard is available in Federal Information Processing Standards Publication 197. The AES algorithm is a symmetric block cipher that can process data blocks of 128 bits, using cipher keys with lengths of 128, 192, and 256 bits.

AES

See Advanced Encryption Standard

attribute

An item of information that describes some aspect of an entry in an LDAP directory. An entry comprises a set of attributes, each of which belongs to an object class. Moreover, each attribute has both a *type*, which describes the kind of information in the attribute, and a *value,* which contains the actual data.

authentication

The process of verifying the identity of a user, device, or other entity in a computer system, often as a prerequisite to granting access to resources in a system. A recipient of an authenticated message can be certain of the message's origin (its sender). Authentication is presumed to preclude the possibility that another party has impersonated the sender.

authentication method

A security method that verifies a user's, client's, or server's identity in distributed environments. Network authentication methods can also provide the benefit of single sign-on

(SSO) for users. The following authentication methods are supported in Oracle Database when Oracle Advanced Security is installed:

- Kerberos
- RADIUS
- Secure Sockets Layer (SSL)
- Windows native authentication

authorization

Permission given to a user, program, or process to access an object or set of objects. In Oracle, authorization is done through the role mechanism. A single person or a group of people can be granted a role or a group of roles. A role, in turn, can be granted other roles. The set of privileges available to an authenticated entity.

auto login wallet

An Oracle Wallet Manager feature that enables PKI- or password-based access to services without providing credentials at the time of access. This auto login access stays in effect until the auto login feature is disabled for that wallet. File system permissions provide the necessary security for auto login wallets. When auto login is enabled for a wallet, it is only available to the operating system user who created that wallet. Sometimes these are called "SSO wallets" because they provide single sign-on capability.

base

The root of a subtree search in an LDAP-compliant directory.

CA

See certificate authority

CDS

See Cell Directory Services (CDS)

Cell Directory Services (CDS)

An external naming method that enables users to use Oracle tools transparently and applications to access Oracle Database databases in a Distributed Computing Environment (DCE).

certificate

An ITU x.509 v3 standard data structure that securely binds an identify to a public key.

A certificate is created when an entity's public key is signed by a trusted identity, a certificate authority. The certificate ensures that the entity's information is correct and that the public key actually belongs to that entity.

A certificate contains the entity's name, identifying information, and public key. It is also likely to contain a serial number, expiration date, and information about the rights, uses, and privileges associated with the certificate. Finally, it contains information about the certificate authority that issued it.

certificate authority

A trusted third party that certifies that other entities—users, databases, administrators, clients, servers—are who they say they are. When it certifies a user, the certificate authority first seeks verification that the user is not on the certificate revocation list (CRL), then verifies the user's identity and grants a certificate, signing it with the certificate authority's private key. The certificate authority has its own certificate and public key which it publishes. Servers and clients use these to verify signatures the certificate authority has made. A certificate authority might be an external company that offers certificate services, or an internal organization such as a corporate MIS department.

certificate chain

An ordered list of certificates containing an end-user or subscriber certificate and its certificate authority certificates.

certificate request

A certificate request, which consists of three parts: certification request information, a signature algorithm identifier, and a digital signature on the certification request information. The certification request information consists of the subject's distinguished name, public key, and an optional set of attributes. The attributes may provide additional information about the subject identity, such as postal address, or a challenge password by which the subject entity may later request certificate revocation. See PKCS #10

certificate revocation lists

(CRLs) Signed data structures that contain a list of revoked certificate **s**. The authenticity and integrity of the CRL is provided by a digital signature appended to it. Usually, the CRL signer is the same entity that signed the issued certificate.

checksumming

A mechanism that computes a value for a message packet, based on the data it contains, and passes it along with the data to authenticate that the data has not been tampered with. The recipient of the data recomputes the cryptographic checksum and compares it with the cryptographic checksum passed with the data; if they match, it is "probabilistic" proof the data was not tampered with during transmission.

Cipher Block Chaining (CBC)

An encryption method that protects against block replay attacks by making the encryption of a cipher block dependent on all blocks that precede it; it is designed to make unauthorized decryption incrementally more difficult. Oracle Advanced Security employs *outer* cipher block chaining because it is more secure than *inner* cipher block chaining, with no material performance penalty.

cipher suite

A set of authentication, encryption, and data integrity algorithms used for exchanging messages between network nodes. During an SSL handshake, for example, the two nodes negotiate to see which cipher suite they will use when transmitting messages back and forth.

cipher suite name

Cipher suites describe the kind of cryptographics protection that is used by connections in a particular session.

ciphertext

Message text that has been encrypted.

cleartext

Unencrypted plain text.

client

A client relies on a service. A client can sometimes be a user, sometimes a process acting on behalf of the user during a database link (sometimes called a proxy).

confidentiality

A function of cryptography. Confidentiality guarantees that only the intended recipient(s) of a message can view the message (decrypt the ciphertext).

connect descriptor

A specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information. The destination service is indicated by using its service name for Oracle9*i* or Oracle8*i* databases or its Oracle system identifier (SID) for Oracle databases version 8.0. The network route provides, at a minimum, the location of the listener through use of a network address. See connect identifier

connect identifier

A connect descriptor or a name that maps to a connect descriptor. A connect identifier can be a net service name, database service name, or net service alias. Users initiate a connect request by passing a username and password along with a connect identifier in a connect string for the service to which they wish to connect:

```
CONNECT username@connect_identifier
Enter password:
```

connect string

Information the user passes to a service to connect, such as username, password and net service name. For example:

```
CONNECT username@net_service_name

Enter password:
```

credentials

A username, password, or certificate used to gain access to the database.

CRL

See certificate revocation lists

CRL Distribution Point

(CRL DP) An optional extension specified by the X.509 version 3 certificate standard, which indicates the location of the Partitioned CRL where revocation information for a certificate is stored. Typically, the value in this extension is in the form of a URL. CRL DPs allow revocation information within a single certificate authority domain to be posted in multiple CRLs. CRL DPs subdivide revocation information into more manageable pieces to avoid proliferating voluminous CRLs, thereby providing performance benefits. For example, a CRL DP is specified in the certificate and can point to a file on a Web server from which that certificate's revocation information can be downloaded.

CRL DP

See CRL Distribution Point

cryptography

The practice of encoding and decoding data, resulting in secure messages.

data dictionary

A set of read-only tables that provide information about a database.

database administrator

(1) A person responsible for operating and maintaining an Oracle Server or a database application. (2) An Oracle username that has been given DBA privileges and can perform database administration functions. Usually the two meanings coincide. Many sites have multiple DBAs. (3) Members of the OracleDBAdmins directory administrative group, who manage the database user-schema mappings for a specific database entry in the directory. Database Configuration Assistant automatically adds the person who registers a database in the directory into the OracleDBAdmins group as the first member of this group for the database being registered.

database alias

See net service name

Database Installation Administrator

Also called a database creator. This administrator is in charge of creating new databases. This includes registering each database in the directory using the Database Configuration Assistant. This administrator has create and modify access to database service objects and attributes. This administrator can also modify the Default domain.

database link

A network object stored in the local database or in the network definition that identifies a remote database, a communication path to that database, and optionally, a username and password. Once defined, the database link is used to access the remote database.

A public or private database link from one database to another is created on the local database by a DBA or user.

A global database link is created automatically from each database to every other database in a network with Oracle Names. Global database links are stored in the network definition.

database method

See Oracle database method

database password verifier

A database password verifier is an irreversible value that is derived from the user's database password. This value is used during password authentication to the database to prove the identity of the connecting user.

Database Security Administrator

The highest level administrator for database enterprise user security. This administrator has permissions on all of the enterprise domains and is responsible for:

- Administering the Oracle DBSecurityAdmins and OracleDBCreators groups.

Creating new enterprise domain**s**.

- Moving databases from one domain to another within the enterprise.

DCE

See Distributed Computing Environment (DCE)

decryption

The process of converting the contents of an encrypted message (ciphertext) back into its original readable format (plaintext).

dictionary attack

A common attack on passwords. the attacker creates a dictionary of many possible passwords and their corresponding verifiers. Through some means, the attacker then obtains the verifier corresponding to the target password, and obtains the target password by looking up the verifier in the dictionary.

digital signature

A digital signature is created when a public key algorithm is used to sign the sender's message with the sender's private key. The digital signature assures that the document is authentic, has not been forged by another entity, has not been altered, and cannot be repudiated by the sender.

directory information tree (DIT)

A hierarchical tree-like structure consisting of the DNs of the entries in an LDAP directory. See distinguished name (DN)

directory naming

A naming method that resolves a database service, net service name, or net service alias to a connect descriptor stored in a central directory server. A

directory naming context

A subtree which is of significance within a directory server. It is usually the top of some organizational subtree. Some directories only permit one such context which is fixed; others permit none to many to be configured by the directory administrator.

Distributed Computing Environment (DCE)

A set of integrated network services that works across multiple systems to provide a distributed environment. The middleware between distributed applications and the operating system or network services; based on a client/server computing model. DCE is supported by the Open Group.

distinguished name (DN)

The unique name of a directory entry. It is comprised of all of the individual names of the parent entries back to the root entry of the directory information tree. See directory information tree (DIT)

domain

Any tree or subtree within the Domain Name System (DNS) namespace. Domain most commonly refers to a group of computers whose host names share a common suffix, the domain name.

Domain Name System (DNS)

A system for naming computers and network services that is organized into a hierarchy of domain**s**. DNS is used in TCP/IP networks to locate computers through user-friendly names. DNS resolves a friendly name into an IP address, which is understood by computers.

In Oracle Net Services, DNS translates the host name in a TCP/IP address into an IP address.

encrypted text

Text that has been encrypted, using an encryption algorithm; the output stream of an encryption process. On its face, it is not readable or decipherable, without first being subject to decryption. Also called ciphertext. Encrypted text ultimately originates as plaintext.

encryption

The process of disguising a message rendering it unreadable to any but the intended recipient.

enterprise domain

A directory construct that consists of a group of databases and enterprise roles. A database should only exist in one enterprise domain at any time. Enterprise domains are different from Windows 2000 domains, which are collections of computers that share a common directory database.

enterprise domain administrator

User authorized to manage a specific enterprise domain, including the authority to add new enterprise domain administrators.

enterprise role

Access privileges assigned to enterprise users. A set of Oracle role-based authorizations across one or more databases in an enterprise domain. Enterprise roles are stored in the directory and contain one or more global roles.

enterprise user

A user defined and managed in a directory. Each enterprise user has a unique identity across an enterprise.

entry

The building block of a directory, it contains information about an object of interest to directory users.

external authentication

Verification of a user identity by a third party authentication service, such as Kerberos or RADIUS.

file system method

Storing fingerprint templates in files when configuring Identix Biometric authentication. The alternative is to use the Oracle database method.

Federal Information Processing Standard (FIPS)

A U.S. government standard that defines security requirements for cryptographic modules—employed within a security system protecting unclassified information within computer and telecommunication systems. Published by the National Institute of Standards and Technology (NIST).

FIPS

See Federal Information Processing Standard (FIPS)

forest

A group of one or more Active Directory trees that trust each other. All trees in a forest share a common schema, configuration, and global catalog. When a forest contains multiple trees, the trees do not form a contiguous namespace. All trees in a given forest trust each other through transitive bidirectional trust relationships.

forwardable ticket-granting ticket

In Kerberos. A service ticket with the `FORWARDABLE` flag set. This flag enables authentication forwarding without requiring the user to enter a password again.

GDS

See Global Directory Service (GDS)

Global Directory Service (GDS)

GDS is the DCE directory service that acts as an agent between DCE CDS and any X.500 directory service. Both GDS and CDS are obsolete; they are only used by DCE.

global role

A role managed in a directory, but its privileges are contained within a single database. A global role is created in a database by using the following syntax:

```
CREATE ROLE <role_name> IDENTIFIED GLOBALLY;
```

grid computing

A computing architecture that coordinates large numbers of servers and storage to act as a single large computer. Oracle Grid Computing creates a flexible, on-demand computing resource for all enterprise computing needs. Applications running on the Oracle 10*g,* or later, grid computing infrastructure can take advantage of common infrastructure services for failover, software provisioning, and management. Oracle Grid Computing analyzes demand for resources and adjusts supply accordingly.

HTTP

Hypertext Transfer Protocol: The set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web. Relative to the TCP/IP suite of protocols (which are the basis for information exchange on the Internet), HTTP is an application protocol.

HTTPS

The use of Secure Sockets Layer (SSL) as a sublayer under the regular HTTP application layer.

identity

The combination of the public key and any other public information for an entity. The public information may include user identification data such as, for example, an e-mail address. A user certified as being the entity it claims to be.

identity management

The creation, management, and use of online, or digital, entities. Identity management involves securely managing the full life cycle of a digital identity from creation (provisioning of digital identities) to maintenance (enforcing organizational policies regarding access to electronic resources), and, finally, to termination.

identity management realm

A subtree in Oracle Internet Directory, including not only an Oracle Context, but also additional subtrees for users and groups, each of which are protected with access control lists.

initial ticket

In Kerberos authentication, an initial ticket or ticket granting ticket (TGT) identifies the user as having the right to ask for additional service tickets. No tickets can be obtained without an initial ticket. An initial ticket is retrieved by running the `okinit` program and providing a password.

instance

Every running Oracle database is associated with an Oracle instance. When a database is started on a database server (regardless of the type of computer), Oracle allocates a memory area called the System Global Area (SGA) and starts an Oracle process. This combination of the SGA and an Oracle process is called an instance. The memory and the process of an instance manage the associated database's data efficiently and serve the one or more users of the database.

integrity

The guarantee that the contents of the message received were not altered from the contents of the original message sent.

java code obfuscation

Java code obfuscation is used to protect Java programs from reverse engineering. A special program (an obfuscator) is used to scramble Java symbols found in the code. The process

leaves the original program structure intact, letting the program run correctly while changing the names of the classes, methods, and variables in order to hide the intended behavior. Although it is possible to decompile and read non-obfuscated Java code, the obfuscated Java code is sufficiently difficult to decompile to satisfy U.S. government export controls.

Java Database Connectivity (JDBC)

An industry-standard Java interface for connecting to a relational database from a Java program, defined by Sun Microsystems.

JDBC

See Java Database Connectivity (JDBC)

KDC

Key Distribution Center. In Kerberos authentication, the KDC maintains a list of user principals and is contacted through the `kinit` (`okinit` is the Oracle version) program for the user's initial ticket. Frequently, the KDC and the Ticket Granting Service are combined into the same entity and are simply referred to as the KDC. The Ticket Granting Service maintains a list of service principals and is contacted when a user wants to authenticate to a server providing such a service. The KDC is a trusted third party that must run on a secure host. It creates ticket-granting tickets and service tickets.

Kerberos

A network authentication service developed under Massachusetts Institute of Technology's Project Athena that strengthens security in distributed environments. Kerberos is a trusted third-party authentication system that relies on shared secrets and assumes that the third party is secure. It provides single sign-on capabilities and database link authentication (MIT Kerberos only) for users, provides centralized password storage, and enhances PC security.

key

When encrypting data, a key is a value which determines the ciphertext that a given algorithm will produce from given plaintext. When decrypting data, a key is a value required to correctly decrypt a ciphertext. A ciphertext is decrypted correctly only if the correct key is supplied.

With a symmetric encryption algorithm, the same key is used for both encryption and decryption of the same data. With an asymmetric encryption algorithm (also called a public-key encryption algorithm or public-key cryptosystem), different keys are used for encryption and decryption of the same data.

key pair

A public key and its associated private key. See public and private key pair

keytab file

A Kerberos key table file containing one or more service keys. Hosts or services use *keytab* files in the same way as users use their passwords.

kinstance

An instantiation or location of a Kerberos authenticated service. This is an arbitrary string, but the host computer name for a service is typically specified.

kservice

An arbitrary name of a Kerberos service object.

LDAP

See Lightweight Directory Access Protocol (LDAP)

ldap.ora file

A file created by Oracle Net Configuration Assistant that contains the following directory server access information:

- Type of directory server
- Location of the directory server
- Default identity management realm or Oracle Context (including ports) that the client or server will use

Lightweight Directory Access Protocol (LDAP)

A standard, extensible directory access protocol. It is a common language that LDAP clients and servers use to communicate. The framework of design conventions supporting industry-standard directory products, such as the Oracle Internet Directory.

listener

A process that resides on the server whose responsibility is to listen for incoming client connection requests and manage the traffic to the server.

Every time a client requests a network session with a server, a listener receives the actual request. If the client information matches the listener information, then the listener grants a connection to the server.

listener.ora file

A configuration file for the listener that identifies the:

- Listener name
- Protocol addresses that it is accepting connection requests on
- Services it is listening for

The `listener.ora` file typically resides in `$ORACLE_HOME/network/admin` on UNIX platforms and `ORACLE_BASE\ORACLE_HOME\network\admin` on Windows.

man-in-the-middle

A security attack characterized by the third-party, surreptitious interception of a message, wherein the third-party, the *man-in-the-middle*, decrypts the message, re-encrypts it (with or without alteration of the original message), and re-transmits it to the originally-intended recipient—all without the knowledge of the legitimate sender and receiver. This type of security attack works only in the absence of authentication.

message authentication code

Also known as data authentication code (DAC). A checksumming with the addition of a secret key. Only someone with the key can verify the cryptographic checksum.

message digest

See checksumming

naming method

The resolution method used by a client application to resolve a connect identifier to a connect descriptor when attempting to connect to a database service.

National Institute of Standards and Technology (NIST)

An agency within the U.S. Department of Commerce responsible for the development of security standards related to the design, acquisition, and implementation of cryptographic-based security systems within computer and telecommunication systems, operated by a Federal agency or by a contractor of a Federal agency or other organization that processes information on behalf of the Federal Government to accomplish a Federal function.

net service alias

An alternative name for a directory naming object in a directory server. A directory server stores net service aliases for any defined net service name or database service. A net service alias entry does not have connect descriptor information. Instead, it only references the location of the object for which it is an alias. When a client requests a

directory lookup of a net service alias, the directory determines that the entry is a net service alias and completes the lookup as if it was actually the entry it is referencing.

net service name

The name used by clients to identify a database server. A net service name is mapped to a port number and protocol. Also known as a connect string, or database alias.

network authentication service

A means for authenticating clients to servers, servers to servers, and users to both clients and servers in distributed environments. A network authentication service is a repository for storing information about users and the services on different servers to which they have access, as well as information about clients and servers on the network. An authentication server can be a physically separate computer, or it can be a facility co-located on another server within the system. To ensure availability, some authentication services may be replicated to avoid a single point of failure.

network listener

A listener on a server that listens for connection requests for one or more databases on one or more protocols. See listener

NIST

See Federal Information Processing Standard (FIPS)

non-repudiation

Incontestable proof of the origin, delivery, submission, or transmission of a message.

obfuscation

A process by which information is scrambled into a non-readable form, such that it is extremely difficult to de-scramble if the algorithm used for scrambling is not known.

obfuscator

A special program used to obfuscate Java source code. See obfuscation

object class

A named group of attribute**s**. When you want to assign attributes to an entry, you do so by assigning to that entry the object classes that hold those attributes. All objects associated with the same object class share the same attributes.

Oracle Context

1. An entry in an LDAP-compliant internet directory called `cn=OracleContext`, under which all Oracle software relevant information is kept, including entries for Oracle Net Services directory naming and checksumming security.

There can be one or more Oracle Contexts in a directory. An Oracle Context is usually located in an identity management realm.

OracleContextAdmins

An administrative group in Oracle Internet Directory whose members have full access to all groups and entries within its associated realm Oracle Context.

Oracle database method

Using an Oracle database to store fingerprint templates when configuring Indentix Biometric authentication. The alternative is to use the file system method.

OracleDBAdmins

An administrative group in Oracle Internet Directory whose members manage the database user-schema mappings for a particular database that is registered in the directory.

OracleDBCreators

An administrative group in Oracle Internet Directory whose members create new databases and registers them in the directory by using Database Configuration Assistant.

OracleDBSecurityAdmins

An administrative group in Oracle Internet Directory whose members have permissions on all of the enterprise domains to configure the identity management realm for enterprise users.

Oracle Net Services

An Oracle product that enables two or more computers that run the Oracle server or Oracle tools such as Designer/2000 to exchange data through a third-party network. Oracle Net Services support distributed processing and distributed database capability. Oracle Net Services is an open system because it is independent of the communication protocol, and users can interface Oracle Net to many network environments.

OraclePasswordAccessibleDomains

See Password-Accessible Domains List

Oracle PKI certificate usages

Defines Oracle application types that a certificate supports.

OracleUserSecurityAdmins

An administrative group in Oracle Internet Directory whose members can administer Oracle database users' security in the directory.

Password-Accessible Domains List

A group of enterprise domain**s** configured to accept connections from password-authenticated users.

PCMCIA cards

Small credit card-sized computing devices that comply with the Personal Computer Memory Card International Association (PCMCIA) standard. These devices, also called PC cards, are used for adding memory, modems, or as hardware security modules. PCMCIA cards used as hardware security modules securely store the private key component of a public and private key pair and some also perform the cryptographic operations as well.

peer identity

SSL connect sessions are between a particular client and a particular server. The identity of the peer may have been established as part of session setup. Peers are identified by X.509 certificate chain**s**.

PEM

The Internet Privacy-Enhanced Mail protocols standard, adopted by the Internet Architecture Board to provide secure electronic mail over the Internet. The PEM protocols provide for encryption, authentication, message integrity, and key management. PEM is an inclusive standard, intended to be compatible with a wide range of key-management approaches, including both symmetric and public-key schemes to encrypt data-encrypting keys. The specifications for PEM come from four Internet Engineering Task Force (IETF) documents: RFCs 1421, 1422, 1423, and 1424.

PKCS #10

An RSA Security, Inc., Public-Key Cryptography Standards (PKCS) specification that describes a syntax for certification requests. A certification request consists of a distinguished name, a public key, and optionally a set of attributes, collectively signed by the entity

requesting certification. Certification requests are referred to as certificate requests in this manual. See certificate request

PKCS #11

An RSA Security, Inc., Public-Key Cryptography Standards (PKCS) specification that defines an application programming interface (API), called Cryptoki, to devices which hold cryptographic information and perform cryptographic operations. See PCMCIA cards

PKCS #12

An RSA Security, Inc., Public-Key Cryptography Standards (PKCS) specification that describes a transfer syntax for storing and transferring personal authentication credentials—typically in a format called a wallet.

PKI

See public key infrastructure (PKI)

plaintext

Message text that has not been encrypted.

principal

A string that uniquely identifies a client or server to which a set of Kerberos credentials is assigned. It generally has three parts: `kservice/kinstance@REALM`. In the case of a user, `kservice` is the username. See also kservice, kinstance, and realm

private key

In public-key cryptography, this key is the secret key. It is primarily used for decryption but is also used for encryption with digital signatures. See public and private key pair

proxy authentication

A process typically employed in an environment with a middle tier such as a firewall, wherein the end user authenticates to the middle tier, which then authenticates to the directory on the user's behalf—as its *proxy*. The middle tier logs into the directory as a *proxy user*. A proxy user can switch identities and, once logged into the directory, switch to the end user's identity. It can perform operations on the end user's behalf, using the authorization appropriate to that particular end user.

public key

In public-key cryptography, this key is made public to all. It is primarily used for encryption but can be used for verifying signatures. See public and private key pair

public key encryption

The process where the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the message is decrypted by the recipient using its private key.

public key infrastructure (PKI)

Information security technology utilizing the principles of public key cryptography. Public key cryptography involves encrypting and decrypting information using a shared public and private key pair. Provides for secure, private communications within a public network.

public and private key pair

A set of two numbers used for encryption and decryption, where one is called the private key and the other is called the public key. Public keys are typically made widely available, while private keys are held by their respective owners. Though mathematically related, it is generally viewed as computationally infeasible to derive the private key from the public key. Public and private keys are used only with asymmetric encryption algorithms, also called public-key encryption algorithms, or public-key cryptosystems. Data encrypted with either a public key or a private key from a key pair can be decrypted with its associated key from the key-pair. However, data encrypted with a public key cannot be decrypted with the same public key, and data enwrapped with a private key cannot be decrypted with the same private key.

RADIUS

Remote Authentication Dial-In User Service (RADIUS) is a client/server protocol and software that enables remote access servers to communication with a central server to authenticate dial-in users and authorize their access to the requested system or service.

realm

1. Short for identity management realm. 2. A Kerberos object. A set of clients and servers operating under a single key distribution center/ticket-granting service (KDC/TGS). Services (see kservice) in different realms that share the same name are unique.

realm Oracle Context

An Oracle Context that is part of an identity management realm in Oracle Internet Directory.

registry

A Windows repository that stores configuration information for a computer.

remote computer

A computer on a network other than the local computer.

root key certificate

See trusted certificate

schema

1. Database schema: A named collection of objects, such as tables, views, clusters, procedures, packages, attributes, object class**es**, and their corresponding matching rules, which are associated with a particular user. 2. LDAP directory schema: The collection of attributes, object classes, and their corresponding matching rules.

schema mapping

See user-schema mapping

Secure Hash Algorithm (SHA)

An algorithm that assures data integrity by generating a 160-bit cryptographic message digest value from given data. If as little as a single bit in the data is modified, the Secure Hash Algorithm checksum for the data changes. Forgery of a given data set in a way that will cause the Secure Hash Algorithm to generate the same result as that for the original data is considered computationally infeasible.

An algorithm that takes a message of less than 264 bits in length and produces a 160-bit message digest. The algorithm with its larger message digest makes it more secure against brute-force collision and inversion attacks.

Secure Sockets Layer (SSL)

An industry standard protocol designed by Netscape Communications Corporation for securing network connections. SSL provides authentication, encryption, and data integrity using public key infrastructure (PKI).

server

A provider of a service.

service

1. A network resource used by clients; for example, an Oracle database server.

2. An executable process installed in the Windows registry and administered by Windows. Once a service is created and started, it can run even when no user is logged on to the computer.

service name

For Kerberos-based authentication, the kservice portion of a service principal.

service principal

See principal

service table

In Kerberos authentication, a service table is a list of service principals that exist on a kinstance. This information must be extracted from Kerberos and copied to the Oracle server computer before Kerberos can be used by Oracle.

service ticket

Trusted information used to authenticate the client. A ticket-granting ticket, which is also known as the initial ticket, is obtained by directly or indirectly running `okinit` and providing a password, and is used by the client to ask for service tickets. A *service ticket* is used by a client to authenticate to a service.

session key

A key shared by at least two parties (usually a client and a server) that is used for data encryption for the duration of a single communication session. Session keys are typically used to encrypt network traffic; a client and a server can negotiate a session key at the beginning of a session, and that key is used to encrypt all network traffic between the parties for that session. If the client and server communicate again in a new session, they negotiate a new session key.

session layer

A network layer that provides the services needed by the presentation layer entities that enable them to organize and synchronize their dialogue and manage their data exchange. This layer establishes, manages, and terminates network sessions between the client and server. An example of a session layer is Network Session.

SHA

See Secure Hash Algorithm (SHA)

shared schema

A database or application schema that can be used by multiple enterprise users. Oracle Advanced Security supports the mapping of multiple enterprise users to the same shared

schema on a database, which lets an administrator avoid creating an account for each user in every database. Instead, the administrator can create a user in one location, the enterprise directory, and map the user to a shared schema that other enterprise users can also map to. Sometimes called user/schema separation.

single key-pair wallet

A PKCS #12-format wallet that contains a single user certificate and its associated private key. The public key is imbedded in the certificate.

single password authentication

The ability of a user to authenticate with multiple databases by using a single password. In the Oracle Advanced Security implementation, the password is stored in an LDAP-compliant directory and protected with encryption and Access Control Lists.

single sign-on (SSO)

The ability of a user to *authenticate once,* combined with strong authentication occurring transparently in subsequent connections to other databases or applications. Single sign-on lets a user access multiple accounts and applications with a single password, entered during a single connection. *Single password, single authentication.* Oracle Advanced Security supports Kerberos, DCE, and SSL-based single sign-on.

smart card

A plastic card (like a credit card) with an embedded integrated circuit for storing information, including such information as user names and passwords, and also for performing computations associated with authentication exchanges. A smart card is read by a hardware device at any client or server.

A smartcard can generate random numbers which can be used as one-time use passwords. In this case, smartcards are synchronized with a service on the server so that the server expects the same password generated by the smart card.

sniffer

Device used to surreptitiously listen to or capture private data traffic from a network.

sqlnet.ora file

A configuration file for the client or server that specifies:

- Client domain to append to unqualified service names or net service names
- Order of naming methods the client should use when resolving a name
- Logging and tracing features to use
- Route of connections

- Preferred Oracle Names servers
- External naming parameters
- Oracle Advanced Security parameters

The `sqlnet.ora` file typically resides in *`$ORACLE_HOME`*`/network/admin` on UNIX platforms and *`ORACLE_BASE`*`\`*`ORACLE_HOME`*`\network\admin` on Windows platforms.

SSO

See single sign-on (SSO)

System Global Area (SGA)

A group of shared memory structures that contain data and control information for an Oracle instance.

system identifier (SID)

A unique name for an Oracle instance. To switch between Oracle databases, users must specify the desired SID. The SID is included in the `CONNECT DATA` parts of the connect descriptor in a tnsnames.ora file, and in the definition of the network listener in a listener.ora file.

ticket

A piece of information that helps identify who the owner is. See service ticket.

tnsnames.ora

A file that contains connect descriptors; each connect descriptor is mapped to a net service name. The file may be maintained centrally or locally, for use by all or individual clients. This file typically resides in the following locations depending on your platform:

- (UNIX) *`ORACLE_HOME`*`/network/admin`
- (Windows) *`ORACLE_BASE`*`\`*`ORACLE_HOME`*`\network\admin`

token card

A device for providing improved ease-of-use for users through several different mechanisms. Some token cards offer one-time passwords that are synchronized with an authentication service. The server can verify the password provided by the token card at any given time by contacting the authentication service. Other token cards operate on a challenge-response basis. In this case, the server offers a challenge (a number) which the user types into the token card. The token card then provides another number (cryptographically-derived from the challenge), which the user then offers to the server.

transport layer

A networking layer that maintains end-to-end reliability through data flow control and error recovery methods. Oracle Net Services uses *Oracle protocol supports* for the transport layer.

trusted certificate

A trusted certificate, sometimes called a root key certificate, is a third party identity that is qualified with a level of trust. The trusted certificate is used when an identity is being validated as the entity it claims to be. Typically, the certificate authorities you trust are called trusted certificates. If there are several levels of trusted certificates, a trusted certificate at a lower level in the certificate chain does not need to have all its higher level certificates reverified.

trusted certificate authority

See certificate authority

trust point

See trusted certificate

username

A name that can connect to and access objects in a database.

user-schema mapping

An LDAP directory entry that contains a pair of values: the base in the directory at which users exist, and the name of the database schema to which they are mapped. The users referenced in the mapping are connected to the specified schema when they connect to the database. User-schema mapping entries can apply only to one database or they can apply to all databases in a domain. See shared schema

user/schema separation

See shared schema

user search base

The node in the LDAP directory under which the user resides.

views

Selective presentations of one or more tables (or other views), showing both their structure and their data.

wallet

A wallet is a data structure used to store and manage security credentials for an individual entity. A Wallet Resource Locator (WRL) provides all the necessary information to locate the wallet.

wallet obfuscation

Wallet obfuscation is used to store and access an Oracle wallet without querying the user for a password prior to access (supports single sign-on (SSO)).

Wallet Resource Locator

A wallet resource locator (WRL) provides all necessary information to locate a wallet. It is a path to an operating system directory that contains a wallet.

Windows native authentication

An authentication method that enables a client single login access to a Windows server and a database running on that server.

WRL

See Wallet Resource Locator

X.509

An industry-standard specification for digital certificate **s**.

Index

A

C

D

E

G

L

M

N

O

P

R

S

V