Oracle® Database

Enterprise User Administrator's Guide

10*g* Release 2 (10.2)

B14269-01

June 2005

ORACLE®

Oracle Database Enterprise User Administrator's Guide 10*g* Release 2 (10.2)

B14269-01

Primary Author: Sumit Jeloka

Contributor: Sarma Namuduri, Srividya Tata, Nina Lewis, Chi Ching Chui, Janaki Narasinghanallur, Min-Hank Ho, Sudha Iyer, Supriya Kalyanasundaram, Lakshmi Kethana, Van Le, Stella Li, Vikram Pesati, Andy Philips, Richard Smith, Deborah Steiner, Philip Thornton, Ramana Turlapati

Contents

2   Configuration and Administration Tools Overview

3 Enterprise User Security Configuration Tasks and Troubleshooting

4 Administering Enterprise User Security

A   Using the User Migration Utility

B SSL External Users Conversion Script

C Integrating Enterprise User Security with Microsoft Active Directory

D Upgrading from Oracle9i to Oracle Database 10*g* Release 2 (10.2)

Glossary

Index

List of Figures

List of Tables

List of Examples

Preface

Welcome to the Oracle Database Enterprise User Administrator's Guide for the 10*g* Release 2 (10.2) 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 Administrator's Guide describes how to implement, configure, and administer Oracle Database users in Oracle Internet Directory, the directory service provided by the Oracle Identity Management platform.

This preface contains these topics:

- Intended Audience
- Documentation Accessibility
- Related Documents
- Conventions

Intended Audience

The Oracle Database Enterprise User 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" on page -xiv.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to

facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

http://www.oracle.com/accessibility/

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, see these Oracle resources:

- *Oracle Internet Directory Administrator's Guide*

- *Oracle Identity Management User Reference Guide*

- *Oracle Database Advanced Security Administrator's Guide*

- *Oracle Database 2 Day DBA*

- *Oracle Database Administrator's Guide*

- *Oracle Database Security Guide*

- *Oracle Database Application Developer's Guide - Fundamentals*

- *Oracle Database SQL Reference*

- *Oracle Database SQL Quick Reference*

- *Oracle Database Error Messages*

- *Oracle Database Reference*

- *Oracle Database Heterogeneous Connectivity Administrator'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.

Printed documentation is available for sale in the Oracle Store at

http://oraclestore.oracle.com/

To download free release notes, installation documentation, white papers, 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

http://www.oracle.com/technology/membership/index.html

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

http://www.oracle.com/technology/documentation/index.html

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

- *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C* by Bruce Schneier. New York: John Wiley & Sons, 1996.

- *SSL & TLS Essentials: Securing the Web* by Stephen A. Thomas. New York: John Wiley & Sons, 2000.

- *Understanding and Deploying LDAP Directory Services* by Timothy A. Howes, Ph.D., Mark C. Smith, and Gordon S. Good . Indianapolis: New Riders Publishing, 1999.

- *Understanding Public-Key Infrastructure: Concepts, Standards, and Deployment Considerations* by Carlisle Adams and Steve Lloyd. Indianapolis: New Riders Publishing, 1999.

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- Conventions in Text
- Conventions in Code Examples
- Conventions for Windows Operating Systems

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle Database Concepts*
		Ensure that the recovery catalog and target database do *not* reside on the same disk.
`UPPERCASE monospace (fixed-width) font`	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a `NUMBER` column.
		You can back up the database by using the `BACKUP` command.
		Query the `TABLE_NAME` column in the `USER_TABLES` data dictionary view.
		Use the `DBMS_STATS.GENERATE_STATS` procedure.

Convention	Meaning	Example
`lowercase monospace (fixed-width) font`	Lowercase monospace typeface indicates executable programs, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names and connect identifiers, user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. *Note:* Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter `sqlplus` to start SQL*Plus. The password is specified in the `orapwd` file. Back up the datafiles and control files in the `/disk1/oracle/dbs` directory. The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table. Set the `QUERY_REWRITE_ENABLED` initialization parameter to `true`. Connect as `oe` user. The `JRepUtil` class implements these methods.
`lowercase italic monospace (fixed-width) font`	Lowercase italic monospace font represents placeholders or variables.	You can specify the *parallel_clause*. Run *old_release*.SQL where *old_release* refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[ ]	Anything enclosed in brackets is optional.	`DECIMAL (digits [ , precision ])`
{ }	Braces are used for grouping items.	`{ENABLE	DISABLE}`
		A vertical bar represents a choice of two options.	`{ENABLE	DISABLE}` `[COMPRESS	NOCOMPRESS]`
...	Ellipsis points mean repetition in syntax descriptions.	`CREATE TABLE ... AS subquery;`
	In addition, ellipsis points can mean an omission in code examples or text.	`SELECT col1, col2, ... , coln FROM employees;`
Other symbols	You must use symbols other than brackets ([ ]), braces ({ }), vertical bars (|), and ellipsis points (...) exactly as shown.	`acctbal NUMBER(11,2);` `acct    CONSTANT NUMBER(4) := 3;`
Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	`CONNECT SYSTEM/system_password` `DB_NAME = database_name`

UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. Because these terms are not case sensitive, you can use them in either UPPERCASE or lowercase.	`SELECT last_name, employee_id FROM employees;`
		`SELECT * FROM USER_TABLES;`
		`DROP TABLE hr.employees;`

Convention	Meaning	Example
lowercase	Lowercase typeface indicates user-defined programmatic elements, such as names of tables, columns, or files.	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;
	Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose **Start** > *menu item*	How to start a program.	To start the Database Configuration Assistant, choose **Start** > **Programs** > **Oracle -** *HOME_NAME* > **Configuration and Migration Tools** > **Database Configuration Assistant**.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (|), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the filename begins with \\, then Windows assumes it uses the Universal Naming Convention.	c:\winnt"\"system32 is the same as C:\WINNT\SYSTEM32
C:\>	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the *command prompt* in this manual.	C:\oracle\oradata>
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	C:\>exp HR/HR TABLES=employees QUERY=\"WHERE job_id='SA_REP' and salary<8000\"
HOME_NAME	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	C:\> net start Oracle*HOME_NAME*TNSListener

Convention	Meaning	Example
ORACLE_HOME and *ORACLE_BASE*	In releases prior to Oracle8*i* release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory. The default for Windows was `C:\orant`. This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle\product\10.1.0`. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is `C:\oracle\product\10.1.0\db_`*n*, where *n* is the latest Oracle home number. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. Refer to *Oracle Database Installation Guide for 32-Bit Windows* for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.	Go to the *ORACLE_BASE**ORACLE_HOME*\rdbms\admin directory.

What's New in Enterprise User Security?

This section describes new features of Enterprise User Security 10*g* Release 2 (10.2) and provides pointers to additional information. New features information from the previous release is also retained to help those users migrating to the current release.

The following sections describe the new features in Enterprise User Security:

- Oracle Database 10g Release 2 (10.2) New Features in Enterprise User Security
- Oracle Database 10g Release 1 (10.1) New Features in Enterprise User Security
- Oracle9i Release 2 (9.2) New Feature in Enterprise User Security

Oracle Database 10*g* Release 2 (10.2) New Features in Enterprise User Security

Enterprise User Security 10*g* Release 2 (10.2) includes the following new features:

- Enterprise User Security 10*g* Release 2 (10.2) includes new functionality for sharing `sqlnet.ora` files among multiple databases. Databases can share a single `sqlnet.ora` file while maintaining separate wallets. This makes Enterprise User Security configuration easier and improves Secure Sockets Layer (SSL) usability. See "Sharing Wallets and sqlnet.ora Files Among Multiple Databases" on page 3-10 for more information.

- Password policies are created for every identity management realm in Oracle Internet Directory. These policies apply to all enterprise users who reside in the realm. Password policies include settings for password complexity, minimum password length, and the like. They also include account lockout and password expiration settings. Enterprise User Security honors the realm wide password policies which are set in Oracle Internet Directory.

  The database communicates with Oracle Internet Directory when authenticating an enterprise user. It checks to see whether the user's account is locked, disabled, expired, or about to expire. It displays appropriate warnings or error messages in these cases.

  > **See Also:** "Password Policies" on page 1-9 for more information on password policies and their management

- The Distinguished Name (DN) in the user certificate no longer needs to match the DN in Oracle Internet Directory. This feature is useful if your Public Key Infrastructure (PKI) certificate authority does not support the use of two common names (cn) in the DN. This also enables you to restructure your Directory without

requiring new certificates for users or databases. See "Configuring Enterprise User Security for SSL Authentication" on page 3-18 for more information.

Enterprise User Security 10*g* Release 2 (10.2) also introduces several new proxying features that enhance both security and ease of use:

- Proxy permissions for specific enterprise users (or lists of enterprise users) can now be created and stored in Oracle Internet Directory. Formerly, proxy permissions could be granted only to a shared schema, necessarily enabling any enterprise user in that shared schema to proxy as the target user.

- Establishing a proxy session results in a single-user session. Formerly, switching from the original connected session to proxy as the target user created a second, independent session, with the first one also remaining active.

- Proxy access is now possible through SQLPLUS as well as Oracle Call Interface (OCI). Formerly, proxy access could be established only through OCI.

New proxying features are described in "Enterprise User Proxy" on page 1-17.

Oracle Database 10*g* Release 1 (10.1) New Features in Enterprise User Security

Enterprise User Security 10g Release 1 (10.1) included the following new features:

- **Kerberos Authenticated Enterprise Users**

  Kerberos-based authentication to the database is available for users managed in an LDAP directory. This includes Oracle Internet Directory or any other third-party directory that is synchronized to work with Oracle Internet Directory by using the Directory Integration Platform. To use this feature, all directory users, including those synchronized from third-party directories, must include the Kerberos principal name attribute (krbPrincipalName attribute).

  > **See Also:** "Configuring Enterprise User Security for Kerberos Authentication" on page 3-16 for configuration details

- **Public Key Infrastructure (PKI) Credentials No Longer Required for Database-to-Oracle Internet Directory Connections**

  In this release, a database can bind to Oracle Internet Directory by using password/SASL-based authentication, eliminating the overhead of setting up PKI credentials for the directory and multiple databases. SASL (Simple Authentication and Security Layer) is a standard defined in the Internet Engineering Task Force RFC 2222. It is a method for adding authentication support to connection-based protocols such as LDAP.

  > **See Also:** "Configuring Enterprise User Security for Password Authentication" on page 3-14 for configuration details

- **Support for User Management in Third-Party LDAP Directories**

  In the current release of Enterprise User Security, you can store and manage your users and their passwords in third-party LDAP directories. This feature is made possible with

  - Directory Integration Platform, which automatically synchronizes third-party directories with Oracle Internet Directory, and

- – Oracle Database recognition of standard password verifiers, which is also new in this release.

- ■ **Tool Changes**

  - – New Tool: Enterprise Security Manager Console

    The Enterprise Security Manager Console, which is based on the Oracle Internet Directory Delegated Administration Service (DAS), is new in this release. Administrators can use this tool to create enterprise users, enterprise user security groups, and to configure identity management realm attributes in the directory that relate to Enterprise User Security.

  - – In this release, Oracle Enterprise Login Assistant functionality has been migrated to the new Enterprise Security Manager Console and Oracle Wallet Manager. The following table lists which tool you should now use to perform tasks that you previously performed by using Oracle Enterprise Login Assistant:

If you used Oracle Enterprise Login Assistant to...	Then now you should use...
Change the directory password	Enterprise Security Manager Console
Change an Oracle wallet password	Oracle Wallet Manager
Enable auto login for an Oracle wallet	Oracle Wallet Manager

> **See Also:** The following sections for information about Enterprise Security Manager Console and how to use it:
>
> - ■ "Enterprise Security Manager Console Overview" on page 2-9, which provides a brief introduction to the tool
>
> - ■ Chapter 4, "Administering Enterprise User Security" that provides procedural information for using the tool to manage enterprise users

Oracle9*i* Release 2 (9.2) New Feature in Enterprise User Security

Enterprise User Security was a feature of Oracle Advanced Security in Oracle9*i* Release 2 (9.2), and it contained the following new feature for that release:

New Tool: User Migration Utility

This utility enables administrators to perform bulk migrations of database users to Oracle Internet Directory for centralized user storage and management.

> **See Also:** Appendix A, "Using the User Migration Utility" for information about this tool and how to use it

1

Getting Started with Oracle Database Enterprise User Security

Enterprise User Security is an important component of Oracle Database 10*g* Release 2 (10.2) Enterprise Edition. 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
- About Using Current User Database Links for Enterprise User Security
- Enterprise User Security Deployment Considerations

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

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, or 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.

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
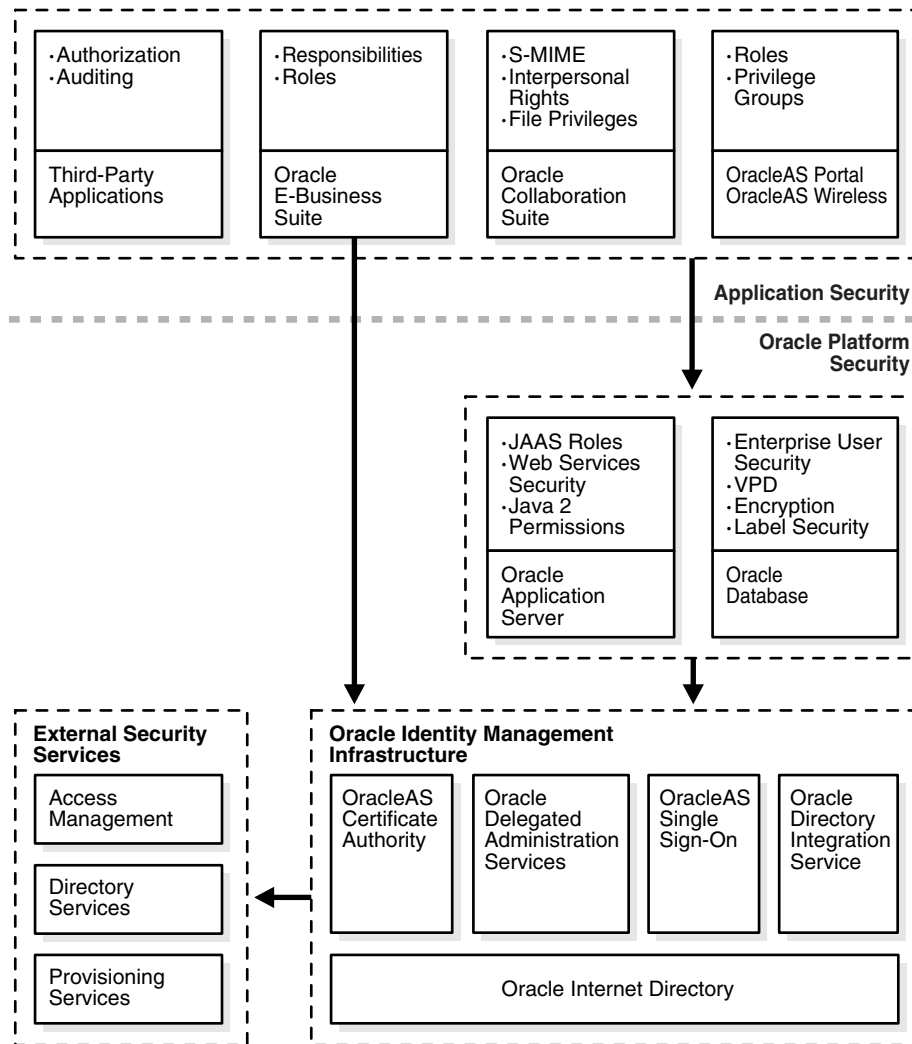
Users benefit from Enterprise User Security through **single sign-on (SSO)** or **single password authentication**, depending on the configuration chosen by the administrator. Using single sign-on, users need to authenticate only once and subsequent authentications take place transparently. This functionality 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 **Access Control Lists (ACLs)**. 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.

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](#)

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

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

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.

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

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 Internet Directory Administrator's Guide* for information about Oracle Internet Directory and its architecture
> - ["About Enterprise User Security Directory Entries"](#) on page 1-8 for information about Oracle Internet Directory entries that are used for Enterprise User Security

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/password@database_service_name
```

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.

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/password@database_service_name
```

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:**
>
> - "Creating New Enterprise Users" on page 4-6
> - *Oracle Database Security Guide* for more information about global users
> - *Oracle Internet Directory Administrator's Guide* for information about defining users in the directory
> - "Enterprise User Proxy" on page 1-17

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.

Private or Exclusive Schemas If users want to retain their individual schemas in the databases that they access, then

- Create enterprise users in the directory, and
- 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.

Shared Schemas To receive the real benefit of the enterprise user solution, you can use shared schemas for your enterprise users. For this strategy

- Create enterprise users in the directory,
- Create a single shared schema in each database, and
- 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" on page 1-14 for more information about creating and using shared schemas for enterprise users
> - "Enterprise User Proxy" on page 1-17

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 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:**
>
> - "About Using Current User Database Links for Enterprise User Security" on page 1-19 for detailed information about creating and using current user database links
>
> - *Oracle Database Administrator's Guide* for information about all of the different types of database links supported by Oracle Database

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*	Supports Oracle8*i* and later clients with Oracle Database 10*g*	Supports Oracle Database 10*g* clients and later with Oracle Database 10*g*
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 10*g* **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:

- Chapter 3, "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

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:** In the Oracle Database 10*g* release, databases must be registered in a complete **identity management realm** of Oracle Internet Directory.

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

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

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) and the Enterprise Security Manager console. 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 that ship with Oracle Application Server 10*g* to create enterprise users in identity management realms. DAS-based tools include Enterprise Security Manager Console and Oracle Internet Directory Self-Service Console.

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

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 his or her 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.

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

- It displays an error if the user account is inactive.

Although the database reads the user account status from Oracle Internet Directory to check for password policy violations, it does not update user account status in Oracle Internet Directory. For example, a failed enterprise user login attempt to the database is not updated in Oracle Internet Directory.

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

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
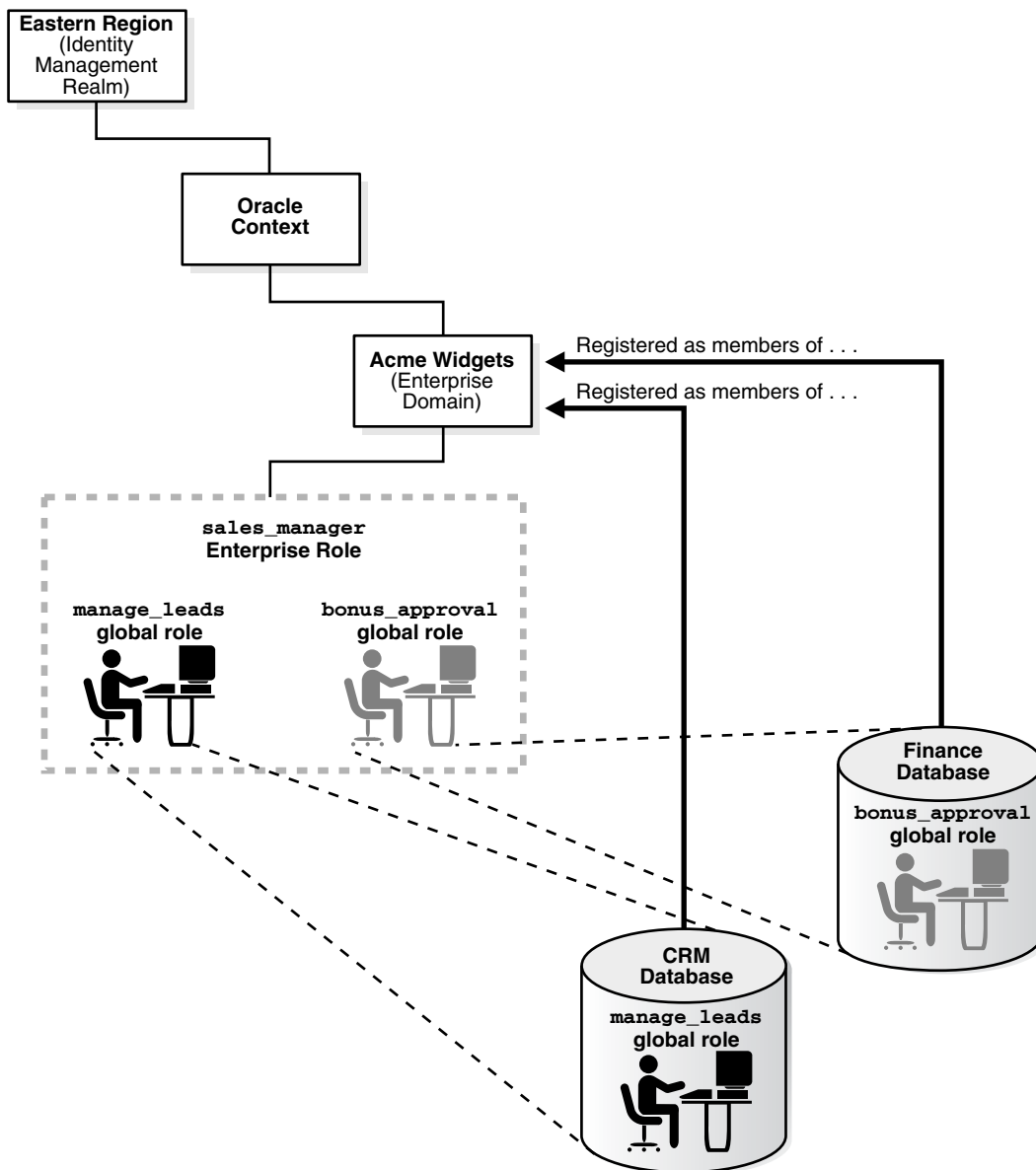
An enterprise role can be assigned to one or more enterprise users. For example, you could assign the enterprise role `sales_manager` to a number of enterprise users who

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 Enterprise Security Manager.

> **See Also:** "Roles: Administering Enterprise Roles" on page 4-23 for information about using Enterprise Security 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.

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 Enterprise Security 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 Enterprise Security Manager.

> **See Also:** "Domains: Administering Enterprise Domains" on page 4-9

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 Enterprise Security Manager.

> **See Also:** "Enterprise Security Manager and Enterprise Security Manager Console" on page 2-3

Database administrators belong to the directory administrative group, **OracleDBAdmins**, which is also managed with the Enterprise Security Manager tool. 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 on page 1-13 for a description of the OracleContextAdmins group
> - "Task 6: Register the database in the directory" on page 3-7
> - "Managing Database Administrators" on page 4-17

Figure 1–3 Related Entries in a Realm Oracle Context
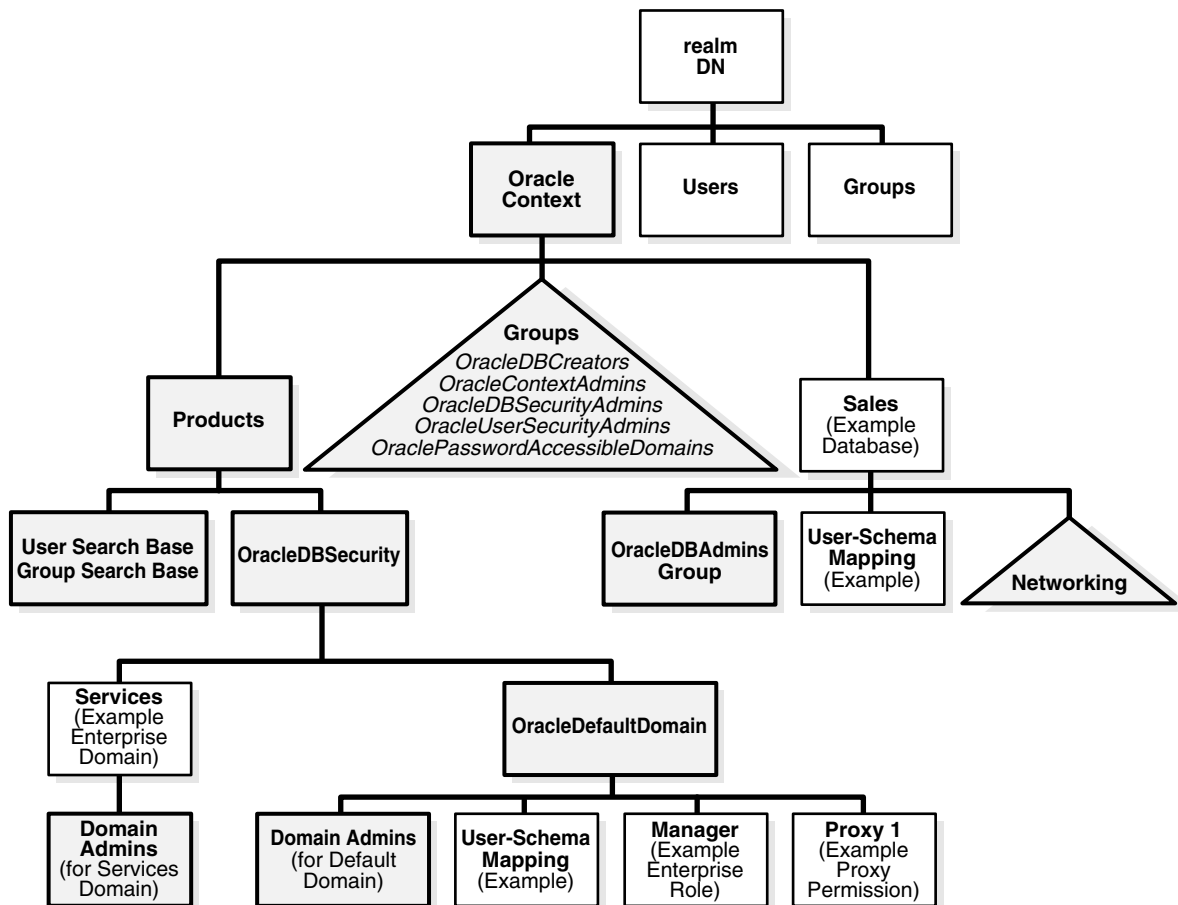
User-Schema Mappings

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" on page 1-16
- "Managing Enterprise Domain Database Schema Mappings" on page 4-13

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 Enterprise Security Manager, Enterprise Security Manager 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 (Called "Full Context Management" group in Release 9.2 and earlier versions of Enterprise Security Manager)	**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 have full access to all groups and entries within the associated realm Oracle Context.
OracleDBAdmins (Accessed by way of the **Administrators** tab for database entries in Enterprise Security Manager)	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 (Called "Database Registration Admins" in Release 9.2 and earlier versions of Enterprise Security Manager)	**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 (Called "Database Security Management" group in Release 9.2 and earlier versions of Enterprise Security Manager)	**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 (Called "Directory User Management" in Release 9.2 and earlier versions of Enterprise Security Manager)	**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.
OraclePasswordAccessible Domains	**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.

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

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.

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 Enterprise Security 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 Enterprise Security 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

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 his or her 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 Enterprise Security 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" on page 3-12 for detailed information about how to create shared schemas for enterprise users

- "Enterprise User Proxy" on page 1-17

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 10*g* Release 2 (10.2) enhances 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 10*g* Release 2 (10.2) 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.

That you can 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 Enterprise Security Manager.

> **See Also:** For more information on configuring enterprise user proxy permissions, see "Managing Proxy Permissions in Enterprise Domains" on page 4-18

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 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 "Managing Proxy Permissions in Enterprise Domains" in Chapter 4.

   > **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]/password@database_service_name
   ```

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

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 as to 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.

About Using Current User Database Links for Enterprise User Security

Oracle Database supports current user database links over an SSL-authenticated network connection. Current user database links let you connect to a second database as yourself, or as another user when used from within a stored procedure owned by that user. Such access is limited to the scope of the procedure. The security advantage of current user database links is that the other user's credentials are not stored in the database link definition and are not sent across the network connection between databases. Instead, security of these links is based on mutual trust, mutual authentication, and a secure network connection between the databases themselves.

For example, a current user database link lets Harriet, a user of the Finance database, procedurally access the Accounts Payable database by connecting as the enterprise user Scott.

For Harriet to access a current user database link to connect to the schema Scott, Scott must be a global schema (created as `IDENTIFIED GLOBALLY`) in both databases. Harriet, however, can be a user identified in one of three ways:

- By a password

- `GLOBALLY`

- `EXTERNALLY`

To create Scott as a global user in the first database, Finance, you must enter

```
CREATE USER Scott IDENTIFIED GLOBALLY as 'CN=Scott,O=nmt'
```

so that Scott has an exclusive schema. Then Scott can map to a shared schema in the second database, Accounts Payable. In order for the current user database link to work, the schema created for Scott in the first database cannot be shared with other users.

Current user database links operate only between trusted databases within a single enterprise domain. Databases within the domain trust each other to authenticate users. You specify an enterprise domain as trusted by using Enterprise Security Manager. When you use Enterprise Security Manager to enable current user database links for a domain, they will work for all databases within that domain. However, each database in the domain must have its own PKI credentials and use SSL to authenticate to the other databases. To specify a database as untrusted that is part of a trusted enterprise domain, use the PL/SQL package `DBMS_DISTRIBUTED_TRUST_ADMIN`. To obtain a list of trusted servers, use the `TRUSTED_SERVERS` view.

> **Note:** Oracle Advanced Security, an option to the Oracle Database Enterprise Edition, does not support RADIUS authentication over database links.

See Also:

- "What Is Meant by Trusted Databases" on page 1-21

- *Oracle Database Heterogeneous Connectivity Administrator's Guide*, for additional information about current user database links

- *Oracle Database SQL Reference,* for more information about SQL syntax

- *PL/SQL Packages and Types Reference*, for information about the PL/SQL package `DBMS_DISTRIBUTED_TRUST_ADMIN`

- *Oracle Database Reference*, for information about the `TRUSTED_SERVERS` view

- *Oracle Database Advanced Security Administrator's Guide*, Chapter 7, for information about configuring SSL for Oracle Net.

- *Oracle Database Advanced Security Administrator's Guide*, Chapter 8, for information about creating wallets

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

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.

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.

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.

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. (Note that Oracle Database supports all verifier types that are supported by Oracle Internet Directory.)

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

SSL provides strong authentication so databases are ensured of each others 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.

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 Enterprise Security Manager. 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 Enterprise Security Manager 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.

> **See Also:**
>
> - "Managing Password-Accessible Domains" on page 4-15
>
> - *Oracle Internet Directory Administrator's Guide* if you are not storing your users in the subtree of an identity management realm. This manual describes how to configure ACLs so password-authenticated users can connect to databases

Considerations for Defining Database Membership in Enterprise Domains

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

- Current user **database link**s operate only between databases within a single **enterprise domain**. Use of these links requires mutual trust between these databases and between the DBAs who administer them.

- 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.

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-Databases (Current User Database Links)	SSL only
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.

Typical Configurations

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

- Password authentication for all connections with no need for current user database links

- SSL authentication for all connections

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

2

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 Enterprise Security 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 Wallet Manager
- Enterprise Security Manager and Enterprise Security Manager Console
- Oracle Net Configuration Assistant
- User Migration Utility
- Duties of an Enterprise User Security Administrator/DBA

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 2–1 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 2–1   Enterprise User Security Tasks and Tools Summary

Task	Tools
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 - Create users and manage their passwords - Manage identity management realm attributes and administrative groups that pertain to Enterprise User Security in Oracle Internet Directory	Enterprise Security Manager and Enterprise Security Manager Console

Table 2–1 (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 (Delegated Administration Service)
Configure databases Oracle home for directory usage over the network	Oracle Net Configuration Assistant
Manage Oracle wallets for Enterprise User Security	Oracle Wallet Manager
Perform bulk migrations of database users to Oracle Internet Directory	User Migration Utility

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.

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 in (or unregister a database from) the directory:" on page 3-8 for information about using this tool to register your database
>
> - *Oracle Database Administrator's Guide* for more information about this tool

Oracle Wallet Manager

Security administrators use Oracle Wallet Manager to manage public key security credentials on Oracle clients and servers. The wallets it creates can be read by Oracle Database, Oracle Application Server 10*g*, and the Oracle Identity Management infrastructure.

> **See Also:** *Oracle Database Advanced Security Administrator's Guide*, the wallet manager chapter

Starting Oracle Wallet Manager

To start Oracle Wallet Manager:

- (Windows) Select **Start**, **Programs**, **Oracle-HOME_NAME**, **Integrated Management Tools**, **Wallet Manager**

- (UNIX) At the command line, enter `owm`.

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 2–2   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:**   Full details of `orapki` commands appear in Appendix E of *Oracle Database Advanced Security Administrator's Guide*

Enterprise Security Manager and Enterprise Security Manager Console

Enterprise User Security employs Enterprise Security Manager and Enterprise Security Manager Console to administer enterprise users, administrative groups, **enterprise domains**, and **enterprise role**s stored in Oracle Internet Directory. (Enterprise Security Manager Console can be accessed through the Enterprise Security Manager **Operations** menu. See "Enterprise Security Manager Console Overview" on page 2-9 for details.)

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:**   Chapter 1, "Getting Started with Oracle Database Enterprise User Security" for a discussion of Enterprise User Security administrative groups, enterprise domains, enterprise roles, enterprise users, shared schemas, and user-schema mappings

This section discusses the following topics:

- Enterprise Security Manager Initial Installation and Configuration Overview
- Starting Enterprise Security Manager
- Navigating the Enterprise Security Manager User Interface
- Enterprise Security Manager Console Overview
- Logging In to Enterprise Security Manager Console
- Navigating Enterprise Security Manager Console User Interface

Enterprise Security Manager Initial Installation and Configuration Overview

Enterprise User Security stores enterprise users in Oracle Internet Directory through Enterprise Security Manager, which requires that you first install and configure Oracle

Internet Directory and OracleAS Single Sign-On. That installation then supports the services needed for enterprise user administration. Oracle Internet Directory provides the Delegated Administration Services, and OracleAS Single Sign-On authenticates administrators when they log in to the administrative GUI (Enterprise Security Manager Console). Oracle Internet Directory and OracleAS Single Sign-On are part of the Oracle Identity Management infrastructure.

The following elements of Oracle Identity Management infrastructure configuration must be completed before proceeding:

- Oracle Internet Directory 10*g* (9.0.4 or later) must be installed, running, and accessible over standard LDAP or Secure Sockets Layer LDAP (LDAP/SSL).

- Oracle Internet Directory must include an identity management realm. You can use Oracle Internet Directory Configuration Assistant to configure this on the directory server.

- OracleAS Single Sign-On server must be installed and configured to authenticate enterprise user security administrators when they log in to the Enterprise Security Manager Console, an element of Enterprise Security Manager.

> **See Also:**
>
> - *Oracle Internet Directory Administrator's Guide* for information about using Oracle Internet Directory Configuration Assistant to create or upgrade an identity management realm in the directory. This manual also contains general information about how to configure and use the directory
>
> - *OracleAS Single Sign-On Administrator's Guide* for information about configuring OracleAS Single Sign-On Server

Once these prerequisite tasks are done, you are ready to install Enterprise Security Manager:

1. Install the Oracle Client.

2. Choose the Administrator installation type.

> **See Also:** The Oracle Database installation documentation for your operating system

> **Note:** Use only the version of Enterprise Security Manager that installs with Oracle Database 10*g* Release 2 (10.2).

Starting Enterprise Security Manager

To launch Enterprise Security Manager, use the following steps:

1. Depending on your operating system, use one of the following options:

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

   - (Windows)

     Choose **Start**, **Programs**, **Oracle - HOME_NAME**, **Integrated Management Tools**, **Enterprise Security Manager**

   The directory server login window appears:

Figure 2–1   Directory Server Login Window
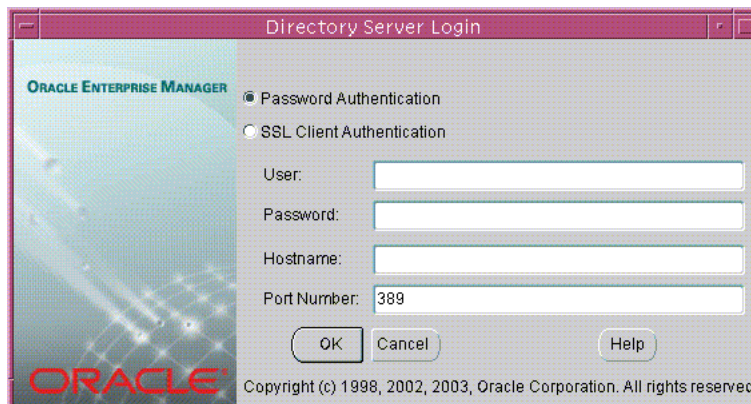
2. To log in to Oracle Internet Directory, supply your full Distinguished Name (DN), select your desired authentication method, and provide the hostname and port number for your directory. Table 2–3 describes the two available Enterprise Security Manager authentication methods and what each method requires:

Table 2–3   Enterprise Security Manager Authentication Methods

Authentication Method	Description
Password Authentication	Uses simple authentication requiring a **distinguished name (DN)** and password[1].
SSL Client Authentication	Uses two-way SSL authentication, in which both the client and server use Oracle Wallets containing digital certificates (that is, the user name and certificate). The subsequent connection is encrypted.

[1]   Known directory user name and password can be used only for the default identity management realm in the directory.

3. After providing the directory login information, click **OK**. The main Enterprise Security Manager user interface appears

Navigating the Enterprise Security Manager User Interface

The Enterprise Security Manager user interface includes two panes, a toolbar, and various menu items as shown in Figure 2–2.

Figure 2–2   Enterprise Security Manager User Interface
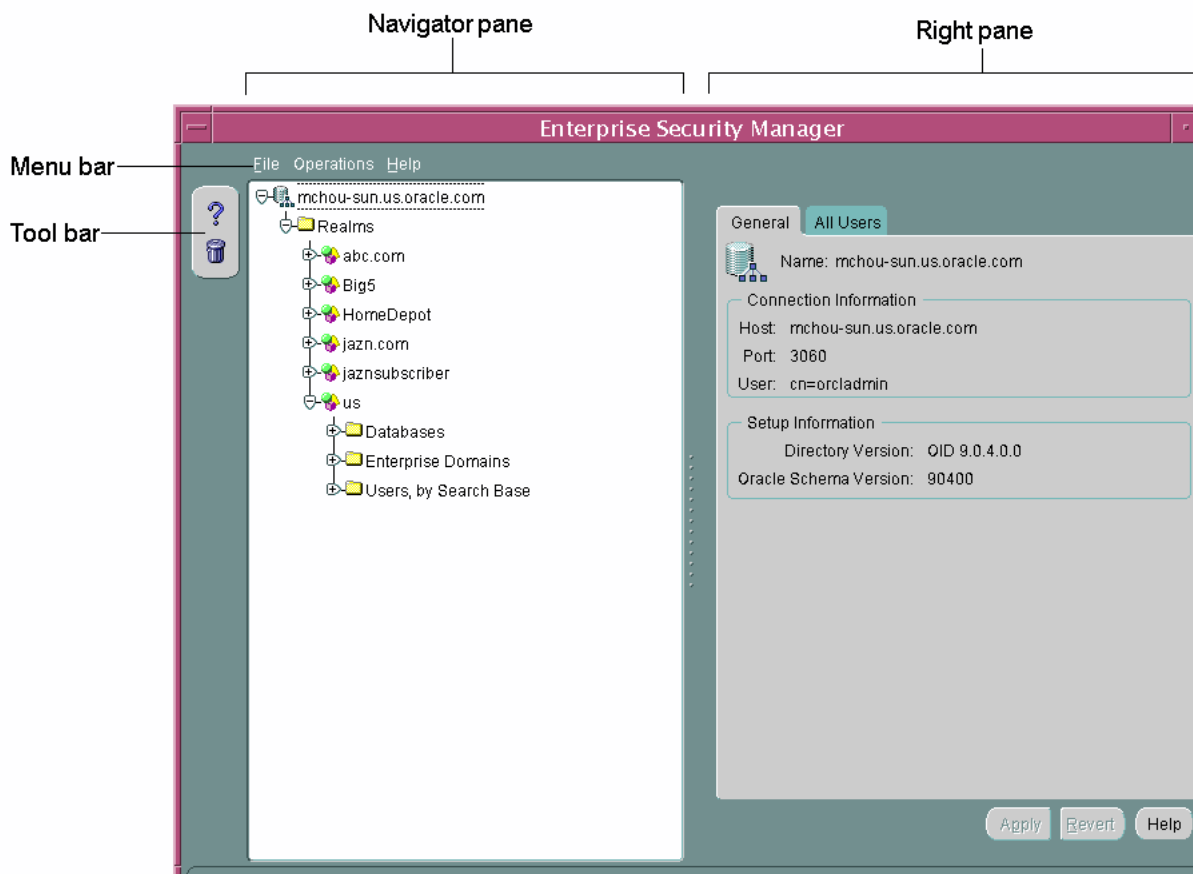
Navigator Pane

The navigator pane provides a graphical tree view of your directory's identity management realms and the databases, enterprise domains, and users they contain. You can use the navigator pane to view, modify, add, or delete enterprise domains and the objects they contain.

The navigator pane enables you to

- Expand and contract identity management realms by clicking the plus (+) and minus (-) symbols adjacent to the realm name in the navigation tree. This enables you to manage the enterprise domains that they contain.

- Right-click an enterprise domain to perform operations such as creating enterprise roles or deleting the domain from the identity management realm.

When you expand an identity management realm, you see a nested list of folders that contain enterprise user security objects. Expanding these folders enables you to view the individual objects as described in Table 2–4.

Table 2–4   Enterprise Security Manager Navigator Pane Folders

Folder	Description: When you expand this folder, you see:
Databases	The list of databases registered with this identity management realm, which was done using Database Configuration Assistant.

Table 2–4   (Cont.) Enterprise Security Manager Navigator Pane Folders

Folder	Description: When you expand this folder, you see:
Enterprise Domains	The enterprise domains that this realm contains. You can also expand each enterprise domain to view the databases and enterprise roles it contains.
Users, by Search Base	The users stored in the realm, organized by *search base*. (User search base is the node in the directory under which a collection of users reside)

Right Pane

The right pane either displays read-only information about the object selected in the navigator pane, or it displays tabbed windows. These windows enable you to configure enterprise domains, enterprise roles, user-schema mappings, and proxy permissions. For example, when you select an enterprise domain in the navigator pane, you can add databases to it by using the Databases tabbed window shown in Figure 2–3.

Figure 2–3   Enterprise Security Manager Databases Tabbed Window
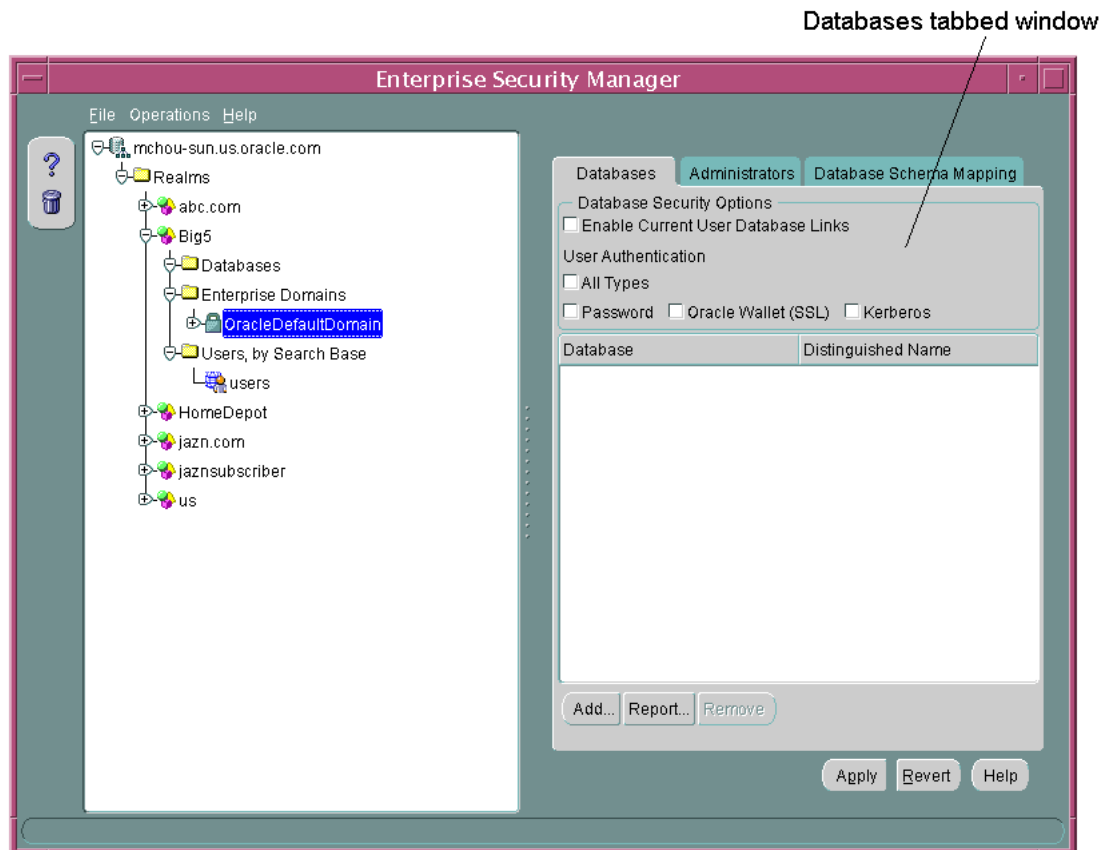
The Databases window also enables you to set security options for databases that are members of an enterprise domain. See "Defining Database Membership of an Enterprise Domain" on page 4-11 for a discussion on configuring enterprise domains by using the Databases tabbed window.

Toolbar

The toolbar on the left of the Enterprise Security Manager user interface has two buttons. The buttons allow you to access the Enterprise Security Manager online help and to delete directory objects.

Menus

You use Enterprise Security Manager menus to create or remove enterprise domains and to manage objects within the domains, such as enterprise roles, database membership, or proxy permissions. The following sections describe the options available under each menu.

File Menu

Table 2–5 describes the contents of the **File** menu.

Table 2–5   Enterprise Security Manager File Menu Options

Option	Description
Change Directory Connection	Causes the Directory Server Login window to reappear (see Figure 2–1 on page 2-5), enabling you to log in to another directory server
Directory Search Options	Enables you to configure user searches in the directory, setting the maximum number of displayed search results, the maximum search duration, or an LDAP filter
ESM Console URL	Enables you to specify the URL for your installation of Enterprise Security Manager Console. (See "Enterprise Security Manager Console Overview" on page 2-9)
Exit	Exits the Enterprise Security Manager application

Operations Menu

Table 2–6 describes the contents of the **Operations** menu.

Table 2–6   Enterprise Security Manager Operations Menu Options

Option	Description
Create Enterprise Domain	Creates an enterprise domain in the realm selected in the navigator pane
Remove Enterprise Domain	Removes the enterprise domain selected in the navigator pane
Create Enterprise Role	Creates an enterprise role in the enterprise domain selected in the navigator pane
Remove Enterprise Role	Removes the enterprise role selected in the navigator pane
Launch ESM Console	Brings up the Enterprise Security Manager Console in your default browser
Create Proxy Permission	Creates a proxy permission in the enterprise domain selected in the navigator pane

Help Menu

Table 2–7 describes the contents of the **Help** menu.

Table 2–7   Enterprise Security Manager Help Menu Options

Option	Description
Contents	Opens the online Help and displays its table of contents
Search for Help on	Displays the search window for the online help
Using Help	Displays online Help topics describing how to use the online Help system
About Enterprise Security Manager	Displays Enterprise Security Manager version number and copyright information

Enterprise Security Manager Console Overview

Enterprise Security Manager uses the directory's Enterprise Security Manager Console to administer enterprise users and groups and to configure an identity management realm for Enterprise User Security. By default, when you log in to a directory server with Enterprise Security Manager, it constructs an Enterprise Security Manager Console URL. (This construction uses port 7777 with the fully qualified domain name of that directory server.) Then, when you need to launch the console, Enterprise Security Manager uses this URL to connect to it over HTTP.

For example, suppose an Acme Company administrator logs in to an instance of Oracle Internet Directory hosted on a machine named `machine123`. Then, Enterprise Security Manager would use the following URL to connect to Enterprise Security Manager Console:

`http://machine123.us.acme.com:7777/`

After launching the console, an administrator must log in by using his or her OracleAS Single Sign-On user name and password pair.

Logging In to Enterprise Security Manager Console

If you can use the default constructed URL to access an instance of Enterprise Security Manager Console, then use the following steps to log in to the console.

To log in to Enterprise Security Manager Console:

1. From the Enterprise Security Manager main application window, choose **Operations**, **Launch ESM Console**.

   The Enterprise Security Manager Console login page appears, as shown in Figure 2–4.

Figure 2–4   Enterprise Security Manager Console Login Page

2. Click the **Login** icon in the upper right corner of the page to log in with your OracleAS Single Sign-On user name and password.

    After providing your OracleAS Single Sign-On credentials, you are returned to the console home page.

To change the default Enterprise Security Manager Console URL:

If you cannot use the default URL to connect to the Enterprise Security Manager Console, then you must enter the appropriate URL before you can launch the console.

1. In the Enterprise Security Manager main application, choose **File**, **ESM Console URL**. The ESM Console URL window appears as shown in Figure 2–5.

Figure 2–5   ESM Console URL Window

1. Enter the appropriate URL for connecting to Enterprise Security Manager Console, and click **OK**.

    This URL information is saved in Enterprise Security Manager, enabling you to launch the console again without reconfiguring the URL.

Configuring Enterprise Security Manager Console for Kerberos-Authenticated Enterprise Users

By default, Enterprise Security Manager 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 window by using the following steps:

1. Log in to the Oracle Internet Directory Self-Service Console and choose the **Configuration** tab. See: *Oracle Internet Directory Administrator's Guide* for information about logging in and using the Oracle Internet Directory Self-Service Console.

2. In the Configuration page, select the **User Entry** subtab and click **Next** until the Configure User Attributes page appears.

3. In the Configure User Attributes page, click **Add New Attribute** and the Add New Attribute page appears.

4. In the Add New Attribute page, select **krbPrincipalName** from the **Directory Attribute Name** list (or the attribute that you have configured for `orclCommonKrbPrincipalAttribute` in your identity management realm) and perform the following steps on this page:

   a. Enter `Kerberos Principal Name` for the user interface label.

   b. Check **Searchable** and **Viewable**.

   c. Select **Single Line Text** from the **UI Type** list.

   d. Click **Done**.

5. Click **Next** to navigate to the Configure Attribute Categories page, and click **Edit** for **Basic Information** and perform the following steps on this page:

   a. Select **krbPrincipalName** in the left category list.

   b. Click **Move,** to move **krbPrincipalName** to the right-hand list.

   c. Click **Done**.

6. Click **Next** until you reach the last page, and then click **Finish** to save your work.

Navigating Enterprise Security Manager Console User Interface

The Enterprise Security Manager Console user interface is browser-based and uses tabbed windows instead of a navigator pane. Figure 2–6 shows the layout of the console user interface. The tabbed windows can be accessed by selecting one of the tabs at the top of the application or by selecting one of the links in the Tips box on the right. You can also access the tabbed windows by selecting one of the corresponding links at the bottom of the page.

Figure 2–6   Enterprise Security Manager Console User Interface
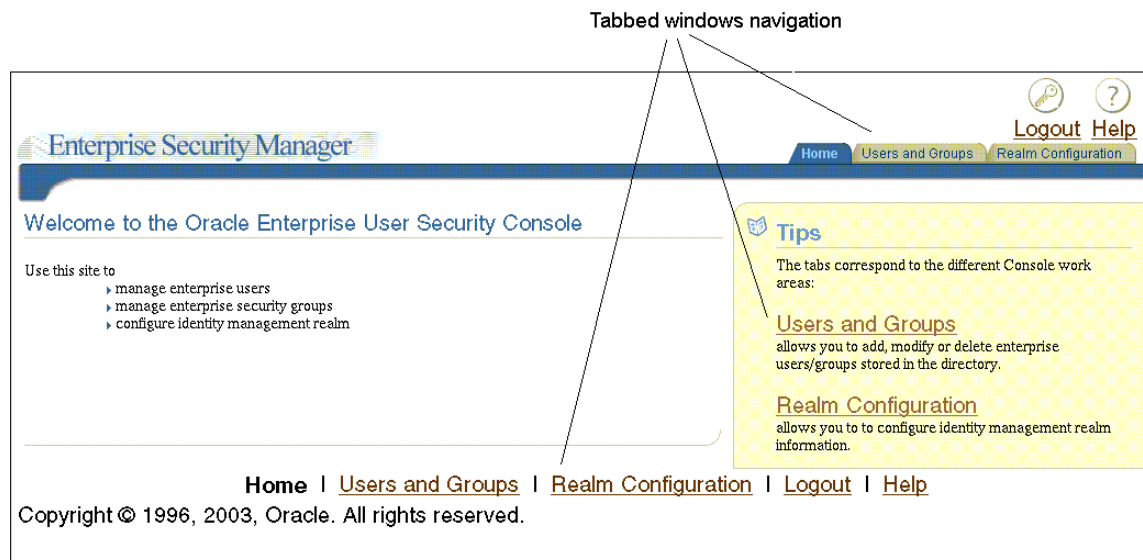
The tabbed windows are explained in the following sections:

Home Tabbed Window

The Home page is your entry point to the console. You can access each tabbed window and read a brief summary of what you can do with this tool. The Home tabbed window is shown in Figure 2–6 on page 2-11.

Users and Groups Tabbed Window

This tabbed window contains two subtabs: the Users subtab (shown in Figure 2–7) and the Groups subtab (shown in Figure 2–8 on page 2-14).

Figure 2–7   Enterprise Security Manager Console Users Subtab

The Users subtab (Figure 2–7) enables you to search for users in the directory by using the **Search for user** field at the top of the page. After you locate users that match your search criteria, you can select specific users and perform tasks with the buttons listed in Table 2–8. This subtab also enables you to create new users.

Table 2–8   Enterprise Security Manager Console User Subtab Buttons

Button Name	Description
Go	After entering user search criteria in the **Search for user** field, click **Go** to display users who match your search criteria in the **Search Results** table. This button is always available.
Create	Enables you to create new enterprise users in the directory. This button is always available.

Table 2–8   (Cont.) Enterprise Security Manager Console User Subtab Buttons

Button Name	Description
Edit	Enables you to edit a user's information in the directory. This button is available only after you have entered search criteria in the **Search for user** field and clicked **Go**.
Delete	Enables you to delete a user from the directory. This button is available only after you have entered search criteria in the **Search for user** field and clicked **Go**.
Assign Privileges	Enables you to assign directory privileges to a specified user. For example, you can assign the privilege to create new users by using this button. This button is available only after you have entered search criteria in the **Search for user** field and clicked **Go**.

The Group subtab (shown in Figure 2–8 on page 2-14) enables you to view or to add new users or groups to the Enterprise User Security directory administrative groups. To view or edit an administrative group, select the adjacent option, and click **Edit** in the upper right corner of the page. When you click **Edit**, an Edit Group page for the specified group appears, displaying the following information:

- Members of the group

- Groups of which the specified administrative group is a member

- Edit history for the group

You can add members or other groups to a specified Enterprise User Security directory administrative group by clicking either **Add User** or **Add Group** in the Member region of the Edit Group page, which is shown in Figure 2–9 on page 2-15.

Figure 2–8   Enterprise Security Manager Console Group Subtab

Figure 2–9   Enterprise Security Manager Console Edit Group Page

Realm Configuration Tabbed Window

The Realm Configuration tabbed window, which is shown in Figure 2–10, enables you to configure identity management realm attributes that pertain to Enterprise User Security. The fields that you can edit on this page are described in Table 2–9 on page 2-16.

Figure 2–10   Enterprise Security Manager Console Realm Configuration Tabbed Window

Table 2–9   Realm Configuration Tabbed Window Fields

Field	Description
Attribute for Login Name	Name of the directory attribute used to store login names
Attribute for Kerberos Principal Name	Name of the directory attribute used to store Kerberos principal names. See also: "Configuring Enterprise Security Manager Console for Kerberos-Authenticated Enterprise Users" on page 2-10
User Search Base	Full **distinguished name (DN)** for the node under which enterprise users are stored for this realm
Group Search Base	Full DN for the node at which user groups (not Enterprise User Security administrative groups) are stored in the directory

Enterprise Security Manager Command-Line Utility

Enterprise Security Manager provides a command-line utility, which can be used to perform the most common tasks that the graphical user interface tool performs. Enter all Enterprise Security Manager command-line utility commands from the Oracle Enterprise Manager Oracle home.

The basic syntax for this utility is as follows:

```
esm -cmd [operation] [-option_1 -option_2 -option_3 ... -option_n]
```

For example, the following command searches for users in a directory installed on a host machine named machine1.us.acme.com:

```
esm -cmd search -U SIMPLE -D orcladmin -w Y4ilbqve -h machine1.us.acme.com
-p 3060 -dn dc=us,dc=acme,dc=com -objectType user
```

The following table describes each option used in this example:

Command Option	Description
-U	Specifies which authentication type used to log in to the directory. SIMPLE specifies password authentication.
-D	Specifies the user name
-w	Specifies the password
-h	Specifies the directory host machine name
-p	Specifies the directory port number
-dn	Specifies the search base
-objectType	Specifies the type of object for which to search

Accessing Enterprise Security Manager Command-Line Utility Help

To view a full list of operations and options you can use with this utility, enter the following at the command line:

```
esm -cmd
```

To view help on a specific operation, enter the following at the command line:

```
esm -cmd help [operation]
```

See Also:

- "Duties of an Enterprise User Security Administrator/DBA" on page 2-19 for a list of tasks that can be performed with Enterprise Security Manager and Enterprise Security Manager Console

- Chapter 4, "Administering Enterprise User Security" for detailed information about how to use Enterprise Security Manager and Enterprise Security Manager Console to administer enterprise users

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. See *Oracle Internet Directory Administrator's Guide* for information about this 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 Internet Directory Administrator's Guide* 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.

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 2–11 on page 2-18.

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 2–11   Opening Page of Oracle Net Configuration Assistant
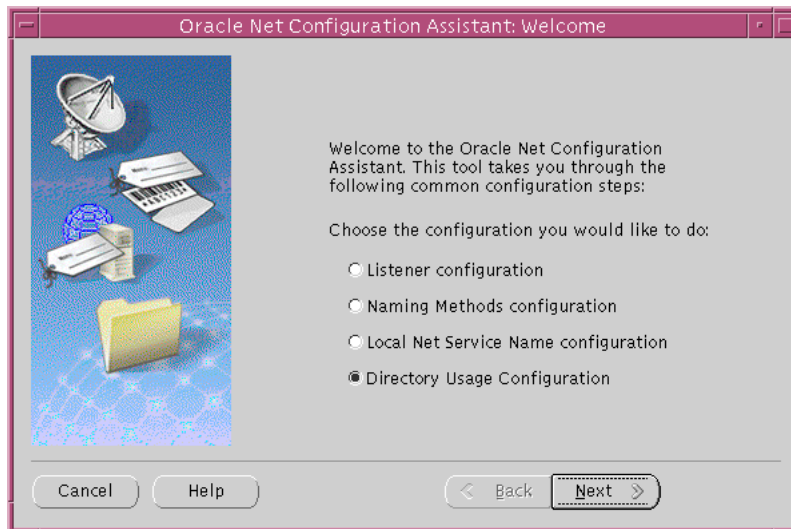
See Also:

- "Task 5: (Optional) Configure your Oracle home for directory usage" on page 3-6 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

User Migration Utility

User Migration Utility is a command-line tool that enables you to perform bulk migrations of database users to Oracle Internet Directory where they are stored and managed as enterprise users. This tool performs a bulk migration in two phases: In phase one, it populates a table with database user information. During phase two, the database user information is migrated to the directory.

This tool is automatically installed in the following location when you install an Oracle Database client:

```
$ORACLE_HOME/rdbms/bin/umu
```

The basic syntax for this utility is as follows:

```
umu parameter_keyword_1=value1:value2
parameter_keyword_2=value
parameter_keyword_3=value1:value2:value3
...
parameter_keyword_n=value
```

Note that when a parameter takes multiple values, they are separated with colons (:).

> **See Also:**   Appendix A, "Using the User Migration Utility" for complete instructions (including usage examples) for using this tool to migrate database users to a directory

Duties of an Enterprise User Security Administrator/DBA

Enterprise User Security administrators plan, implement, and administer enterprise users. Table 2–10 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 2–10   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 Internet Directory Administrator's Guide* for information about how to perform this task
Upgrade an identity management realm in Oracle Internet Directory	Oracle Internet Directory Configuration Assistant	*Oracle Internet Directory Administrator's Guide* 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 Internet Directory Administrator's Guide* (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" on page 3-6
Register a database in the directory	Database Configuration Assistant	"Task 6: Register the database in the directory" on page 3-7
Configure password authentication for Enterprise User Security	■ Enterprise Security Manager ■ Oracle Net Manager	"Configuring Enterprise User Security for Password Authentication" on page 3-14
Configure Kerberos authentication for Enterprise User Security	■ Oracle Net Manager ■ Enterprise Security Manager Console ■ Enterprise Security Manager	"Configuring Enterprise User Security for Kerberos Authentication" on page 3-16
Configure SSL authentication for Enterprise User Security	■ Oracle Net Manager ■ Enterprise Security Manager ■ Oracle Wallet Manager	"Configuring Enterprise User Security for SSL Authentication" on page 3-18
Create or modify user entries and Oracle administrative groups in the directory	Enterprise Security Manager Console	■ "Realms: Administering Realms in Identity Management" on page 4-2 ■ "Users: Administering Enterprise Users" on page 4-5
Create or modify enterprise roles and domains in the directory	Enterprise Security Manager	■ "Domains: Administering Enterprise Domains" on page 4-9 ■ "Roles: Administering Enterprise Roles" on page 4-23
Create or modify wallets for directory, databases, and clients	■ Oracle Wallet Manager ■ orapki command line utility	*Oracle Database Advanced Security Administrator's Guide*: ■ Oracle Wallet Manager chapter ■ orapki appendix
Change a user's database or directory password	Enterprise Security Manager Console	"Setting Enterprise User Passwords" on page 4-7
Change a database's directory password	Database Configuration Assistant	"To change the database's directory password:" on page 3-9
Manage user wallets on the local system or update database and directory wallet passwords	Oracle Wallet Manager	*Oracle Database Advanced Security Administrator's Guide*
Request initial Kerberos ticket when KDC is not part of the operating system, such as Kerberos V5 from MIT	`okinit` utility	*Oracle Database Advanced Security Administrator's Guide* for information about using the `okinit` utility to get an initial Kerberos ticket
Migrate large numbers of local or external database users to the directory for Enterprise User Security	User Migration Utility	Appendix A, "Using the User Migration Utility"

3

Enterprise User Security Configuration Tasks and Troubleshooting

This chapter describes configuring Enterprise User Security by 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:

- Enterprise User Security Configuration Overview
- Enterprise User Security Configuration Roadmap
- Preparing the Directory for Enterprise User Security (Phase One)
- Configuring Enterprise User Security Objects in the Database and the Directory (Phase Two)
- Configure Enterprise User Security for the Authentication Method You Require (Phase Three)
- Enabling Current User Database Links
- Troubleshooting Enterprise User Security

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
- Database-to-database (current user database links can be secured by SSL only)

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 3–1 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 3–1   Enterprise User Security Configuration Flow Chart

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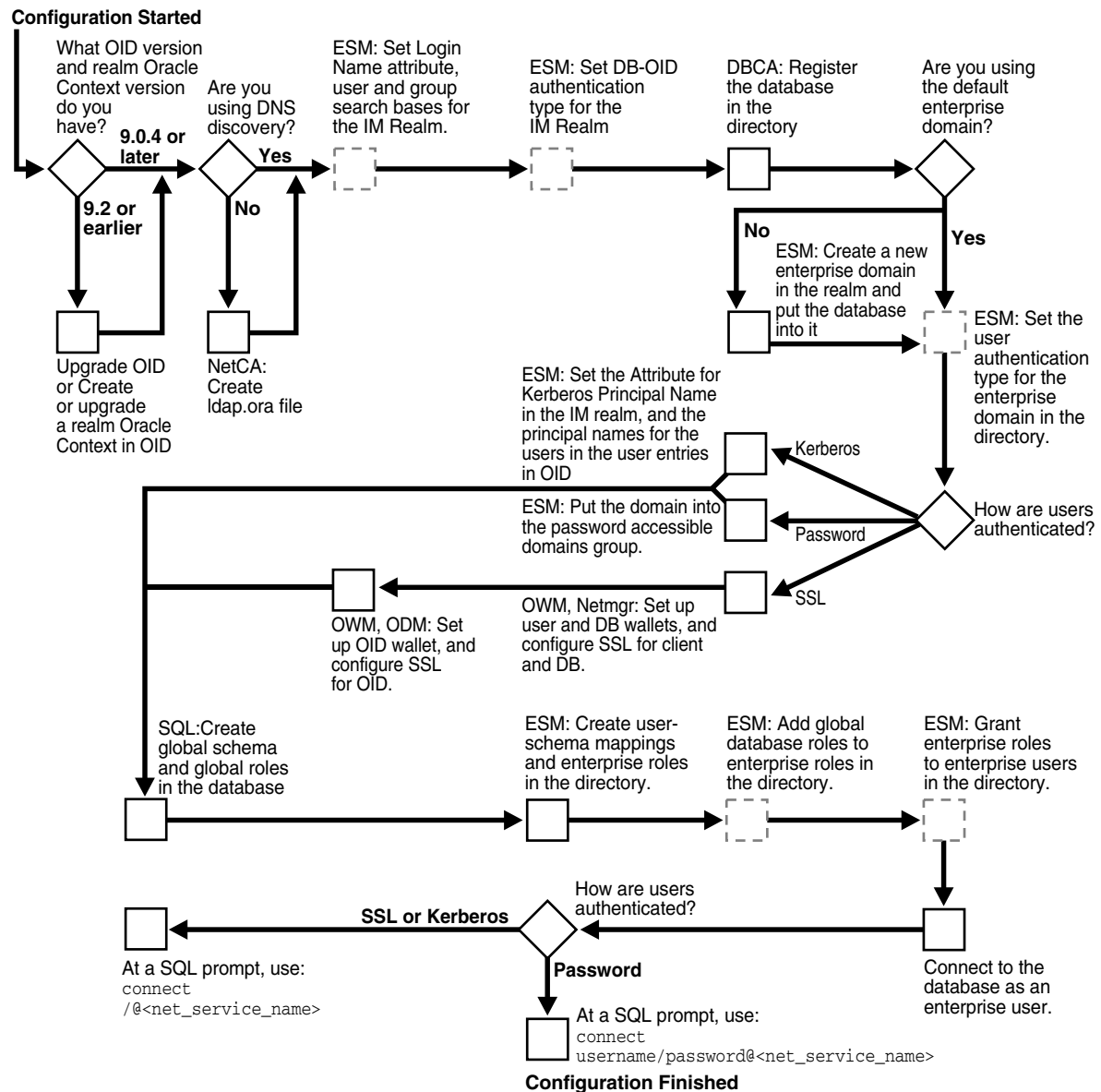
For brevity, some product names and features have been abbreviated in this flow chart. The following table lists the abbreviations used and the meaning of each:

Abbreviation	Meaning
DBCA	Database Configuration Assistant
ESM	Enterprise Security Manager
IM Realm	Identity Management Realm
Netmgr	Oracle Net Manager
ODM	Oracle Directory Manager
OID	Oracle Internet Directory
OWM	Oracle Wallet Manager
SQL	SQL*Plus

> **See Also:** Chapter 1, "Getting Started with Oracle Database Enterprise User Security" for information about the realm Oracle Context, its administrative groups, and entries that pertain to Enterprise User Security

Enterprise User Security Configuration Roadmap

This section provides detailed descriptions of the configuration steps that Figure 3–1 illustrates. They should be performed in the following order:

1. "Preparing the Directory for Enterprise User Security (Phase One)" on page 3-4

2. "Configuring Enterprise User Security Objects in the Database and the Directory (Phase Two)" on page 3-11

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" on page 3-14

   – "Configuring Enterprise User Security for Kerberos Authentication" on page 3-16

   – "Configuring Enterprise User Security for SSL Authentication" on page 3-18

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 10g Release 1 (10.1) 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 10*g* Release 2 (10.2). Version 9.0.4 realms are backward compatible to Oracle9*i*, so you can register Oracle9*i* and Oracle Database 10*g* Release 2 (10.2) in the same realm and place them in the same domain, if desired.

Task 2: (Optional) Set identity management realm properties

Table 3–1 shows the defaults for a version 9.0.4 identity management realm.

Table 3–1   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 Enterprise Security Manager 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" on page 4-3.

> **Note:**
>
> - To manage enterprise domains and enterprise roles in the directory, use only the versions of Enterprise Security Manager that are current with your Oracle Database release. For example, to manage enterprise domains and roles for Oracle Database 10*g* Release 2 (10.2), use only Enterprise Security Manager 10*g* Release 2 (10.2). Alternatively, if you are using an Oracle9*i* Database, use only the Oracle9*i* Enterprise Security Manager to administer enterprise roles and domains for that database.
>
> - 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 Chapter 4, "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 Enterprise Security Manager 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" on page 4-4.

> **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 Internet Directory Administrator's Guide* 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 Internet Directory Administrator's Guide* for information about DNS server discovery.)

If you are *not* using DNS discovery, then you *must* use Oracle Net Configuration Assistant 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" on page 2-17)

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 Internet Directory Administrator's Guide*.)

- 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.

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.oracle.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 wallet that Database Configuration Assistant automatically generates during database registration can only be used with an Oracle Database 10*g* Release 2 (10.2) instance.
  >
  > - You cannot use this database wallet for earlier versions of the database. Also, this wallet cannot be use by Oracle Internet Directory Release 9.0.4 or earlier to start an SSL server.
  >
  > - The database's password-based credentials for authentication to Oracle Internet Directory are placed in the wallet when an Oracle Database 10*g* is registered in Oracle Internet Directory.
  >
  > - You cannot use earlier versions of Oracle Wallet Manager to manage Oracle Database 10*g* Release 2 (10.2) wallets that contain these 10*g* Release 2 (10.2) credentials.

To register a database in (or unregister a database from) the directory:

See "Starting Database Configuration Assistant" on page 2-2 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 the directory credentials for a user in the OracleDBCreators group.

5. 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" on page 3-10 for further information.

6. Click **Finish** if you are only registering the database. Click **Next** if you want to configure additional database features.

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" on page 2-2 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 Enterprise Security Manager to remove it from the enterprise domain.

- Does not remove the database wallet.

> **See Also:** To delete the wallet, see the discussions on Oracle Wallet Manager in *Oracle Database Advanced Security Administrator's Guide*. The particular sections of note are named "Navigating the Oracle Wallet Manager User Interface" in Chapter 2 and "Managing Wallets" in Chapter 9

> **Note:** To succeed at unregistering an Oracle Database 10*g* Release 2 (10.2) 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).

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 Oracle Wallet Manager. However, 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.

> **See Also:** *Oracle Database Advanced Security Administrator's Guide* for information about using Oracle Wallet Manager to change wallet passwords and, in general, to manage public key infrastructure (PKI) credentials

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)" on page 3-11.

> **See Also:**
>
> - *Oracle Internet Directory Administrator's Guide* 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

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 amongst 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.

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)" on page 3-4.

- 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 Internet Directory Administrator's Guide* 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 a New Enterprise Domain" on page 4-10 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 Reference* for information about the syntax used for these steps

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

Use Enterprise Security Manager (see "Starting Enterprise Security Manager" on page 2-4) to configure user-schema mappings for the OracleDefaultDomain by using the following steps:

1. Select the **OracleDefaultDomain** in the navigator pane.

2. Select the Database Schema Mapping tabbed window and click **Add**.

3. In the Add Database Schema Mappings dialog box enter the appropriate DN and the shared schema name that you created in Task 1. Refer to the Enterprise Security Manager online Help for information about how to enter these values.

4. Click **OK**. The new user-schema mappings apply to all databases in the enterprise domain.

For more information about this task, see "Managing Enterprise Domain Database Schema Mappings" on page 4-13.

> **Note:** You also can create user-schema mappings under a database in an enterprise domain which apply only to that database.

Task 3: Create Enterprise Roles in the Enterprise Domain

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

1. Right-click OracleDefaultDomain in the navigator pane and choose **Create Enterprise Role**.

   The Create Enterprise Role dialog box appears with the appropriate realm Oracle Context and enterprise domain displayed.

2. Enter the enterprise role name in the **Role Name** field.

3. Click **OK**. The new enterprise role is added under the domain in the navigator pane.

For more information about this task, see "Creating a New Enterprise Role" on page 4-23.

Task 4: Add Global Database Roles to Enterprise Roles

Use Enterprise Security 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 name in the navigator pane.

2. Choose the Database Global Roles tabbed window and click **Add**.

3. In the Add Global Database Roles dialog box, select the database from which to obtain global roles. A database logon window appears, prompting you for a user name and password to authenticate to the database so global roles can be fetched. Typically, this is a DBA logon to the database.

   > **Note:** You can use the database name that appears by default in the Service field to connect to the database if your Oracle home has **LDAP** as one of its selected Oracle Net naming methods, or if this name appears as a TNS alias in your local Oracle Net configuration. Otherwise, you can overwrite the Service field with any other TNS alias (from the database tnsnames.ora file), or by using a connect string in the following format:
   >
   > <host>:<port>:<oracle_SID>
   >
   > For example: machine111:1521:sales_db

4. Click **OK**. Enterprise Security Manager connects to the selected database, fetches the global roles supported on that database, and displays them in the Add Global Database Roles dialog box.

5. Select one or more global roles and click **OK**. The selected global roles appear in the Database Global Roles window.

6. Click **Apply**. The new global roles are added to the enterprise role.

For more information about this task, see "Assigning Database Global Role Membership to an Enterprise Role" on page 4-24.

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

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

1. In the navigator pane, select an enterprise role in the appropriate identity management realm.

2. Select the Users tab adjacent to the main application window and click **Add**.

3. In the Add Enterprise Users dialog box top panel, select a directory entry as a user search base, or edit the Selection field to manually define the user search base.

4. In the middle Search Criteria panel, check **Include Subtrees** to enable searching for all users within the search, including subtrees.

5. Enter any known user name in the **Show Names Containing** field. This limits the search to users in the directory who have a common name value that contains or starts with the specified text.

6. Click **Search Now**. If there are any users in the directory who match your search criteria, then they are listed in the bottom panel.

7. Choose a desired user by selecting the user in the bottom panel and clicking **OK**, or by double-clicking the user. Multiple users can be granted the enterprise role by selecting a range of users and clicking **OK**.

   The Add Enterprise Users dialog box automatically closes, and you are returned to the main application window.

8. The user names you added appear in the Users tab. Click **Apply** to grant the enterprise role to the users.

For more information about this task, see "Granting Enterprise Roles to Users" on page 4-26.

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" on page 3-14

- "Configuring Enterprise User Security for Kerberos Authentication" on page 3-16

- "Configuring Enterprise User Security for SSL Authentication" on page 3-18

  > **See Also:** Table 1–1, " Enterprise User Security Authentication: Selection Criteria" on page 1-7 for a comparison of the benefits provided by password, Kerberos, and SSL authentication for Enterprise User Security

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)" on page 3-4.

- 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)" on page 3-11.

- You have configured an SSL instance with no authentication for Oracle Internet Directory as described in *Oracle Internet Directory Administrator's Guide*. 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: Add the Enterprise Domain to the Password-Accessible Domains List

- Task 3: 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 Enterprise Security Manager to enable password authentication for OracleDefaultDomain and add it to Password-Accessible Domains List by using the following steps:

1. Select **OracleDefaultDomain** in the navigator pane.

2. Choose the Databases tabbed window and select **Password** or **All Types** from the **User Authentication** methods listed.

3. Click **Apply**.

4. Select the identity management realm in the navigator pane.

5. Choose the Accessible Domains tabbed window and click **Add**.

6. In the Add Accessible Enterprise Domains dialog box, select **OracleDefaultDomain** from the list of enterprise domains, and click **OK**. OracleDefaultDomain is added to the password-accessible domains list.

For more information about this task, see Table 4–4, " Enterprise Security Manager Database Security Options" on page 4-12.

Task 2: Add the Enterprise Domain to the Password-Accessible Domains List

Use Enterprise Security Manager to add OracleDefaultDomain to Password-Accessible Domains List by performing the following steps:

1. Select the identity management realm in the navigator pane.

2. Choose the Accessible Domains tabbed window and click **Add**.

3. In the Add Accessible Enterprise Domains dialog box, select **OracleDefaultDomain** from the list of enterprise domains, and click **OK**. OracleDefaultDomain is added to the Password-Accessible Domains List.

For more information about this task, see "Managing Password-Accessible Domains" on page 4-15.

Task 3: Connect as a Password-Authenticated Enterprise User

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

```
SQL> connect hscortea/welcome@<Oracle Net Service Name>
```

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-# Errors for Password-Authenticated Enterprise Users" on page 3-23.

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" on page 3-29.

You have completed password-authenticated Enterprise User Security configuration.

> **See Also:**
>
> - "Troubleshooting Enterprise User Security" on page 3-23 for information about diagnosing and resolving errors
>
> - Chapter 4, "Administering Enterprise User Security" for information about configuring the identity management realm, and about creating and managing enterprise domains, enterprise roles, and enterprise users

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 the Kerberos chapter of the *Oracle Database Advanced Security Administrator's Guide*.

- You have prepared your directory by completing the tasks described in "Preparing the Directory for Enterprise User Security (Phase One)" on page 3-4.

- 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)" on page 3-11.

- You have configured an SSL instance with no authentication for Oracle Internet Directory as described in *Oracle Internet Directory Administrator's Guide*. 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 Enterprise Security Manager 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 Enterprise Security Manager Console to display the Kerberos principal name attribute

Use Oracle Internet Directory Self-Service Console to configure the Enterprise Security Manager Console to display the Kerberos principal name attribute. For more information about this task, see "Configuring Enterprise Security Manager Console for Kerberos-Authenticated Enterprise Users" on page 2-10.

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

Use Enterprise Security Manager 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" on page 4-3.

> **Note:** By default, the Enterprise Security Manager 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. See "Configuring Enterprise Security Manager Console for Kerberos-Authenticated Enterprise Users" on page 2-10 for details.

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

Use Enterprise Security Manager 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" on page 4-6.

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 Enterprise Security Manager to enable Kerberos authentication for your enterprise domain by performing the following steps:

1. Select the enterprise domain in the navigator pane.

2. Choose the Databases tabbed window and select **Kerberos** or **All Types** from the **User Authentication** methods listed.

3. Click **Apply**.

For more information about this task, see "Managing Database Security Options for an Enterprise Domain" on page 4-12.

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 "Obtaining the Initial Ticket with the okinit Utility" in the *Oracle Database Advanced Security Administrator's Guide*.

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-# Errors for Kerberos-Authenticated Enterprise Users" on page 3-26.

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" on page 3-29.

You have completed Kerberos-authenticated Enterprise User Security configuration.

> **See Also:**
>
> - "Troubleshooting Enterprise User Security" on page 3-23 for information about diagnosing and resolving errors
>
> - Chapter 4, "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

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 Oracle Wallet Manager 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 his or her 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 his or her 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 3–2 describes the values of the `orclpkimatchingrule` attribute.

Table 3–2   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.

> **See Also:** *Oracle Internet Directory Administrator's Guide* and *Oracle Identity Management User Reference Guide* 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 "Enabling SSL" in the *Oracle Database Advanced Security Administrator's Guide*. 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 Internet Directory Administrator's Guide*.

- You have prepared your directory by completing the tasks described in "Preparing the Directory for Enterprise User Security (Phase One)" on page 3-4.

- 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)" on page 3-11.

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

Use Enterprise Security Manager to enable SSL authentication for the **enterprise domain** (OracleDefaultDomain) by using the following steps:

1. Select the enterprise domain in the navigator pane.

2. Choose the Databases tabbed window and select **Oracle Wallet (SSL)** or **All Types** from the **User Authentication** methods listed.

3. Click **Apply**.

For more information about this task, see "Managing Database Security Options for an Enterprise Domain" on page 4-12.

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 Administrator's Guide* for information about editing initialization parameters
>
> - *Oracle Database Reference* for information about the `LDAP_DIRECTORY_ACCESS` initialization parameter
>
> - *Oracle Database SQL 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.

> **See Also:** *Oracle Database Advanced Security Administrator's Guide* for the default location of a user's wallet when the authentication used between the user and the database is SSL

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

1. Use Oracle Wallet Manager to download a user wallet from the directory. See "Downloading a Wallet from an LDAP Directory" in the Oracle Wallet Manager chapter of *Oracle Database Advanced Security Administrator's Guide*.

2. Use Oracle Wallet Manager to enable autologin for the user wallet. Enabling autologin generates a single sign-on (`.sso`) file and enables authentication to the SSL adapter. See *Oracle Database Advanced Security Administrator's Guide* for information about using the autologin feature of Oracle Wallet Manager.

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-# Errors for SSL-Authenticated Enterprise Users" on page 3-28.

   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" on page 3-29.

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. See *Oracle Database Advanced Security Administrator's Guide* for information about using Oracle Wallet Manager to disable auto login for an Oracle wallet.

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.oracle.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" on page 3-23 for information about diagnosing and resolving errors
>
> - Chapter 4, "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

Enabling Current User Database Links

Current user database links require SSL-enabled network connections between the databases. Before you can enable current user database links, you must enable SSL, create Oracle wallets, and obtain PKI credentials for all databases involved.

Then, use Enterprise Security Manager to enable current user database links between databases within the **enterprise domain** in the directory by using the following steps:

1. Select the enterprise domain in the navigator pane.

2. Choose the Databases tabbed window and check **Enable Current User Database Links**.

3. Click **Apply**.

For more information about this task, see "Managing Database Security Options for an Enterprise Domain" on page 4-12.

Troubleshooting Enterprise User Security

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

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

ORA-# Errors for Password-Authenticated Enterprise Users

If you receive an ORA-# 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" on page 3-29

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 Internet Directory Administrator's Guide* 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 Oracle Wallet Manager 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>" -w <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. In 10*g* Release 2 (10.2), Oracle Database supports only the `viewEntry` mode.

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 Enterprise Security Manager 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 Enterprise Security Manager 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 Enterprise Security Manager 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 Enterprise Security Manager 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 Enterprise Security Manager Console or 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 Enterprise Security Manager 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. Use Enterprise Security Manager to check that the enterprise domain is in the password-accessible domains group.

5. 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 Enterprise Security Manager 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" on page 3-30

ORA-28862: SSL connection failed

    Cause: As in error message

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

ORA-# Errors for Kerberos-Authenticated Enterprise Users

If you receive an ORA-# 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" on page 3-29

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" on page 3-23 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" on page 3-24 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 Enterprise Security Manager to set the user authentication policy for this enterprise domain to **KERBEROS** or **ALL**.

    2. See "DOMAIN-READ-ERROR Checklist" on page 3-30

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 Enterprise Security Manager Console or 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 Enterprise Security Manager 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 Enterprise Security Manager 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 Enterprise Security Manager 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>"
-w <password>
```

> **Note:** The `mkstore` utility is for troubleshooting purposes only. The name and functionality of this tool may change in the future. In 10*g* Release 2 (10.2), Oracle Database supports only the `viewEntry` mode.

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.

ORA-# Errors for SSL-Authenticated Enterprise Users

If you receive an ORA-# 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" on page 3-29

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 Internet Directory Administrator's Guide* 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 Oracle Wallet Manager 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>" -P <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. In 10*g* Release 2 (10.2), Oracle supports only the `viewEntry` mode.

ORA-28301: Domain policy has not been registered for SSL authentication

Cause: As in error message

Action: Use Enterprise Security 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 Advanced Security Administrator's Guide* for information about configuring your SSL connection.

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 Enterprise Security Manager to check that the global role is included in an enterprise role in the directory.

3. Use Enterprise Security 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" on page 3-30.

USER-SCHEMA ERROR Checklist

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

1. 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.

2. 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>';
```

3. 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 Oracle Wallet Manager 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>';
   ```

4. If you are using a shared schema, then check the following:

   - Use Enterprise Security 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>" -P <wallet_password> -b "<database_DN>"
       "objectclass=*"
       ```

       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> -w
       <database_directory_password> -b "<database_DN>" "objectclass=*"
       ```

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

     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" on page 3-30.

DOMAIN-READ-ERROR Checklist

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

1. Use Enterprise Security 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>" -P <wallet_password> -b "cn=OracleContext,
   <realm_DN>" "objectclass=orclDBEnterpriseDomain"
   ```

   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> -w
   <database_directory_password> -b "cn=OracleContext, <realm_DN>"
   "objectclass=orclDBEnterpriseDomain"
   ```

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

   This `ldapsearch` should return exactly one enterprise domain.

   If no domain is returned and Enterprise Security 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 Enterprise Security 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>" -P <wallet_password> -b "cn=OracleContext,
   <realm_DN>" "objectclass=orclDBEnterpriseRole"
   ```

   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> -w
   <database_directory_password> -b "cn=OracleContext, <realm_DN>"
   "objectclass=orclDBEnterpriseRole"
   ```

   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 Enterprise Security Manager to create enterprise roles and mappings.

4. Use Enterprise Security Manager to set or reset the user authentication policy for the relevant enterprise domain. See "Managing Database Security Options for an

Enterprise Domain" on page 4-12 for information about setting the user authentication policy for an enterprise domain.

4

Administering Enterprise User Security

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

- Enterprise User Security Administration Tools Overview
- Realms: Administering Realms in Identity Management
- Users: Administering Enterprise Users
- Domains: Administering Enterprise Domains
- Roles: Administering Enterprise Roles

Enterprise User Security Administration Tools Overview

Enterprise Security Manager and Enterprise Security Manager Console are the two
main tools provided for administering Enterprise User Security.

Use Enterprise Security Manager to create and manage

- Enterprise domains
- Enterprise roles

Use Enterprise Security Manager Console to create, manage, and configure

- Enterprise users
- Enterprise User Security administrative groups
- Identity management realm properties.

These tools are introduced in Chapter 2, "Configuration and Administration Tools
Overview" where you can find information about starting each tool and navigating its
interface.

In particular, refer to the following topics to get started using Enterprise User Security
administration tools:

Tool	Introductory Topics
Enterprise Security Manager	"Enterprise Security Manager and Enterprise Security Manager Console" on page 2-3 "Enterprise Security Manager Initial Installation and Configuration Overview" on page 2-3 "Starting Enterprise Security Manager" on page 2-4

Tool	Introductory Topics
Enterprise Security Manager Console	■ "Enterprise Security Manager Console Overview" on page 2-9 ■ "Logging In to Enterprise Security Manager Console" on page 2-9 ■ "Navigating Enterprise Security Manager Console User Interface" on page 2-11

Realms: Administering Realms in Identity Management

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. Enterprise Security Manager is one such product. It lets you manage database and security-related information in an identity management realm.

This section describes how to use Enterprise Security Manager to administer directory identity management realm properties that pertain to Enterprise User Security. It contains the following topics:

■ Identity Management Realm Versions

■ Setting Properties of an Identity Management Realm

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

  ■ 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" on page 1-4 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" on page 1-8 for a discussion on the Oracle Internet Directory entries that are used for Enterprise User Security

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). You can manage Enterprise User Security directory entries in a version 9.0.4 (or later) identity management realm by using Enterprise Security Manager for Oracle Database 10*g* Release 2 (10.2).

Enterprise Security Manager displays all existing version 9.0.4 (or later) identity management realms in its main application tree.

> **Note:** Enterprise User Security did not require identity management realms in Oracle8*i*, nor in Oracle9*i*. In those previous releases, only an Oracle Context was used. For Oracle Database 10*g* Release 2 (10.2) Enterprise User Security, full identity management realms and their associated realm Oracle Contexts must be used.

Setting Properties of an Identity Management Realm

An identity management realm has a number of properties that can be viewed and managed by using Enterprise Security Manager. These properties are described in Table 4–1.

Table 4–1   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 10*g* Release 1 (10.1) or *10g Release 2 (10.2)* 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.

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. Navigate to the Enterprise Security Manager Console home page. (Select **Launch Enterprise Security Manager Console** from the Operations menu and log in by using your OracleAS Single Sign-On user name and password.)

2. Select the **Realm Configuration** tab.

3. In the Realm Information window, enter the appropriate information into the available fields.

4. Click **Submit** to save your changes to the directory.

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.

To set the default database-to-directory authentication type for an identity management realm:

1. Select the identity management realm in the left navigator pane.

2. Select the **General** tab in the right main window. See Figure 4–1.

Figure 4–1   Enterprise Security Manager: General Tab
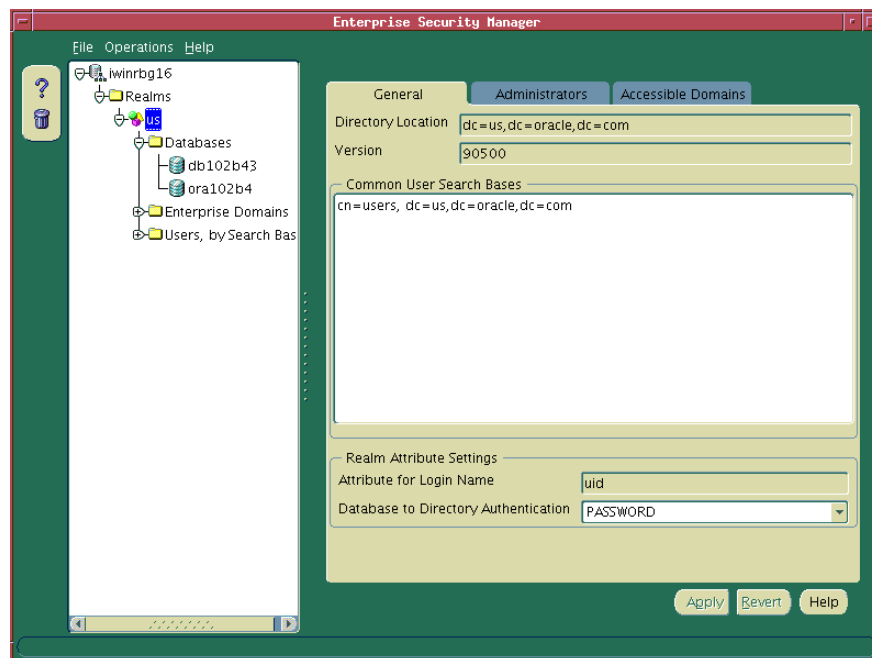
3. In the Realm Attribute Settings region of the General tabbed window, select either **PASSWORD** or **SSL** from the **Database to Directory Authentication** list.

4. Click **Apply** to save your changes to the directory.

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 4–2. For more information about these groups, see "Administrative Groups" on page 1-13.

Table 4–2   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. Navigate to the Enterprise Security Manager Console home page. (Select **Launch Enterprise Security Manager Console** from the Operations menu and log in by using your OracleAS Single Sign-On user name and password.)

2. Select the **Users and Groups** tab.

3. In the Users and Groups tabbed window, select the **Group** subtab.

4. In the Group subtab window, select the administrative group you wish to edit, and click **Edit**.

5. In the Edit Group window, enter group information into the appropriate fields. You can change group owners, add users to or remove them from groups, and view group membership.

6. Click **Submit** to save your changes to the directory.

Users: Administering Enterprise Users

Enterprise Security Manager manages one directory server at a time, identified at the top of the main application tree. It lets you manage enterprise users and data that is relevant to Enterprise User Security in the identity management realm.

This section describes how to use Enterprise Security Manager to administer enterprise users. It contains the following topics:

- Creating New Enterprise Users

- Setting Enterprise User Passwords

- Defining an Initial Enterprise Role Assignment

- Browsing Users in the Directory

Creating New Enterprise Users

Use Enterprise Security Manager 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" on page 4-3

To create new enterprise users:

1. Select **Launch Enterprise Security Manager Console** from the Operations menu. The Enterprise Security Manager Console home page appears (Figure 4–2). Log in with your OracleAS Single Sign-On user name and password.

Figure 4–2 Enterprise Security Manager Console Home Page

2. Select the Users and Groups tab.

3. In the Users and Groups tabbed window, choose the User subtab, if it is not already displayed.

4. In the User subtab window, click **Create** (located on the upper right corner of the Search Results table). Note that if your users are authenticated to the database by using Kerberos credentials, and the `krbPrincipalName` attribute is not there, then see "Configuring Enterprise Security Manager Console for Kerberos-Authenticated Enterprise Users" on page 2-10 for information about how to configure this.

5. Enter the appropriate user information in the Create User window and click **Submit** to create a new enterprise user.

Setting Enterprise User Passwords

You can set and maintain enterprise user passwords in the Basic Information region of the Enterprise Security Manager Console Edit User window (Figure 4–3).

Figure 4–3   Enterprise Security Manager Console Edit User Window: Basic Information

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. Navigate to the Enterprise Security Manager Console home page. (Select **Launch Enterprise Security Manager Console** from the Operations menu and log in using your OracleAS Single Sign-On user name and password.)

2. Select the **Users and Groups** tab.

3. In the Users and Groups tabbed window, select the **User** subtab, if it is not already displayed.

4. In the User subtab window, enter part of the enterprise user's user name (login name) or e-mail address, and click **Go**.

   A list of all users who match your search criteria displays.

5. Select the user for whom you wish to create a new password, and click **Edit**.

6. In the Edit User window, enter the new password, and click **Submit**.

Defining an Initial Enterprise Role Assignment

When you create a new enterprise user, you can grant any previously configured enterprise roles to the new user.

> **See Also:** "Roles: Administering Enterprise Roles" on page 4-23

To assign existing enterprise roles to a new enterprise user:

1. In the left navigator pane, select the **Users** icon under the Users, By Search Base folder, which is displayed under the identity management realm you are using. The list of users is displayed in the right main window.

2. Select a user in the main window, and click **Edit**. An Edit User window displays.

3. Select the Enterprise Roles tab of the Edit User window, and click **Add**.

   The Add Enterprise Roles window appears (Figure 4–4):

Figure 4–4   Enterprise Security Manager: Add Enterprise Roles Window
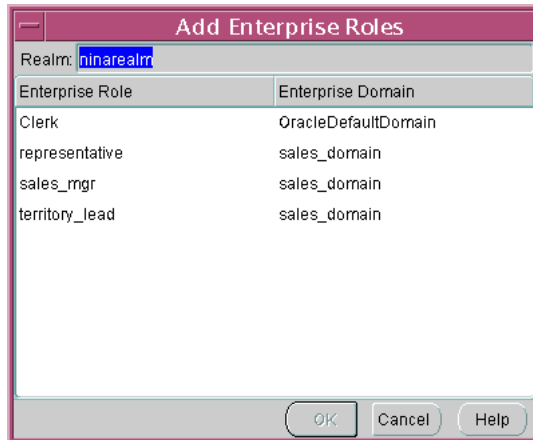
4. Select the correct identity management realm, then select any enterprise roles in your realm to assign to the new user, and click **OK**.

Browsing Users in the Directory

Enterprise Security Manager lets you browse the directory for all users currently stored there in two ways, by using Enterprise Security Manager Console or by using the All Users tab in the main application window.

To browse enterprise users in the directory by using Enterprise Security Manager Console:

1. Navigate to the Enterprise Security Manager Console home page. (Select **Launch Enterprise Security Manager Console** from the Operations menu and log in by using your OracleAS Single Sign-On user name and password.)

2. Select the **Users and Groups** tab.

3. In the Users and Groups tabbed window, select the **User** subtab, if it is not already displayed.

4. In the User subtab window, enter part of the enterprise user's user name (login name) or e-mail address, and click **Go**. To display all users, do not enter search criteria.

   A list of all users that match your search criteria is displayed. You can browse through the displayed users and select one to edit, delete, or assign privileges. If you need to create a new user, then click **Create**.

To browse enterprise users in the directory by using the All Users tab in the main application window:

1. Select the directory in the left navigator pane.

2. Select the **All Users** tab in the right main window (Figure 4–5):

Figure 4–5   Enterprise Security Manager: Main Window (All Users Tab)
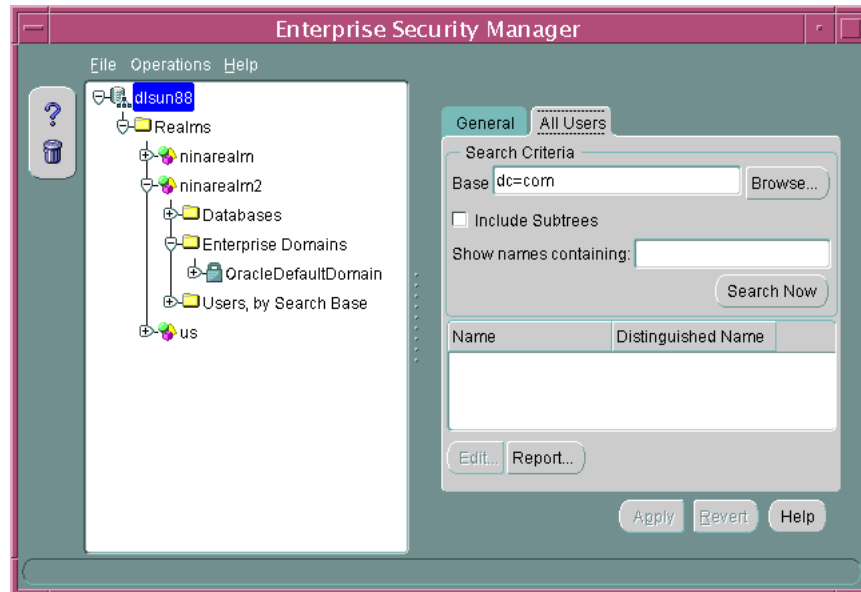
3. Define the search criteria and click **Search Now**. The window displays the results of the search. Table 4–3 summarizes the search criteria and their respective effects on the search results.

Table 4–3   Directory Search Criteria

Search Criteria	Effect on the Search
Base	This is the base entry point in the directory where the search is performed. Only users under this base are returned by the search.
Include Subtrees	This determines whether to show *all* users found in the entire subtree under the selected base or to show only those users who exist directly under that base location (one level only).
Show names containing	This *limits the search* to those users whose directory entries have a common name that starts with the characters you specify. This is useful if you do not know the exact name or base of the target users.

Note that you can also browse enterprise users in the directory by selecting **<realm_ name>**, **User by Search Base**, **Users** in the left navigation tree of the main application window.

Domains: Administering Enterprise Domains

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 Enterprise Security Manager to administer enterprise domains in the directory. It contains the following topics:

- Creating a New Enterprise Domain
- Defining Database Membership of an Enterprise Domain
- Managing Database Security Options for an Enterprise Domain
- Managing Enterprise Domain Administrators
- Managing Enterprise Domain Database Schema Mappings
- Managing Password-Accessible Domains
- Managing Database Administrators
- Managing Proxy Permissions in Enterprise Domains

Creating a New Enterprise Domain

If you do not want to use `OracleDefaultDomain`, then you can create a new enterprise domain in your identity management realm.

To create a new enterprise domain in an identity management realm:

1. Start by using one of the following methods:

   - Select **Create Enterprise Domain** from the Operations menu.

   - The Create Enterprise Domain window appears (Figure 4–6):

Figure 4–6   Enterprise Security Manager: Create Enterprise Domain Window

   - In the **Create Enterprise Domain** window, select the appropriate realm from the list (Figure 4–6).

   > **Note:**   If you invoked the Create Enterprise Domain window by right-clicking the realm in the main application tree, then the name of that realm is already selected.

2. Enter the name of the new enterprise domain, in the **Domain Name** field.

3. Click **OK**. The new enterprise domain is created in the realm and appears on the main application tree.

To remove an enterprise domain:

1. Select the target enterprise domain from the main application tree.

2. Select **Remove Enterprise Domain** from the Operations menu.

3. Enterprise Security Manager prompts you to confirm removal of the enterprise domain from the realm. Click **OK** to remove it.

> **Note:** You cannot remove an enterprise domain from an identity management realm if that enterprise domain contains any enterprise roles.

Defining Database Membership of an Enterprise Domain

Use the navigation tree of the main Enterprise Security Manager window to select a specific enterprise domain. You can then use the Databases tab to manage database membership of an enterprise domain in a realm (Figure 4–7):

Figure 4–7   Enterprise Security Manager: Databases Tab (Database Membership)
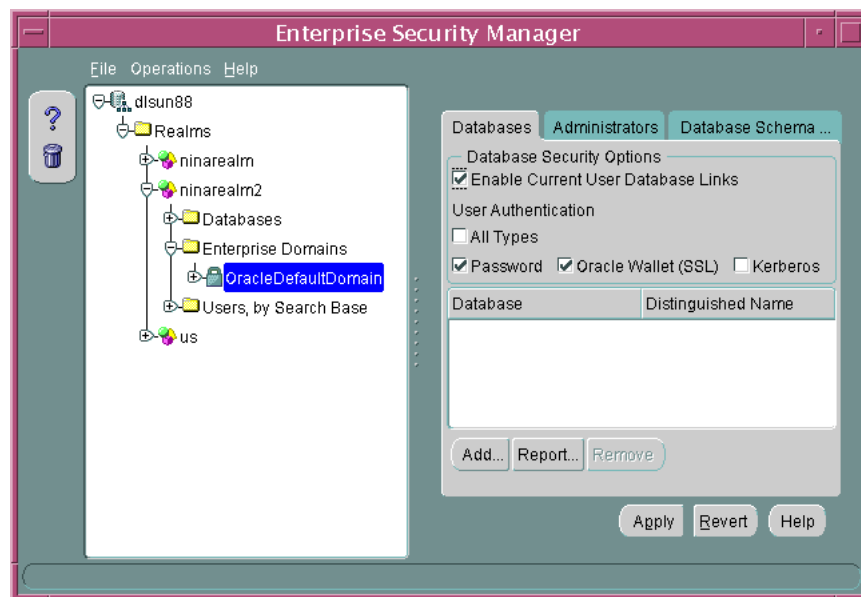
To remove a database from an enterprise domain:

1. Select a specific database for removal, and click **Remove**. The database is removed from the list.

2. Click **Apply**. The database is removed from the enterprise domain.

To add a database to an enterprise domain:

> **Note:** The following restrictions apply to adding databases to an enterprise domain:
>
> ■ A database must be in an enterprise domain for enterprise users to be able to connect to it.
>
> ■ 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.

1. Click **Add**. The Add Databases window appears. This window lists all the databases that are not part of any domain associated with the realm(Figure 4–8):

Figure 4–8   Enterprise Security Manager: Add Databases Window
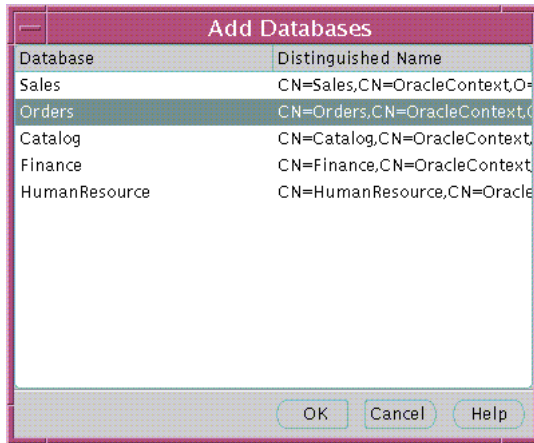
2. Select a new database to be added to the enterprise domain.

3. Click **OK**. The selected database is added to the list of databases in the Databases tabbed window (Figure 4–7).

4. Click **Apply** (Figure 4–7). The new database is added to the enterprise domain.

Managing Database Security Options for an Enterprise Domain

Use the Databases tabbed window (Figure 4–7) to manage database security options applicable to all databases that are members of the enterprise domain.

Database security options are summarized in Table 4–4:

Table 4–4   Enterprise Security Manager Database Security Options

Database Security Option	Description
Enable current user database links	Any database pair can permit use of *Current User Database Links* only if both databases exist in the same enterprise domain where this setting is enabled. By default, current user database links are not enabled.
User authentication	All databases in an enterprise domain allow one, or more, of the following types of authentication for clients: ■ All (the default setting) Databases can accept all currently available authentication methods for Enterprise User Security. In 10*g* Release 2 (10.2), this includes passwords, SSL by using PKI credentials, or Kerberos credentials. ■ Password ■ SSL (PKI certificates) ■ Kerberos

Managing Enterprise Domain Administrators

An **enterprise domain administrator** is a directory user with privileges to modify the content of that domain. After selecting an enterprise domain under an realm in the

main application tree, you can use the Administrators tabbed window to manage Enterprise Domain Administrators.

To add a new user to the list of Enterprise Domain Administrators:

1. In the left navigator pane, select the enterprise domain to which you wish to add administrators.

2. In the right pane, select the **Administrators** tab.

3. Click **Add**. The Add Users window appears. Use this window to locate and select users for designation as Enterprise Domain Administrators. The new users appear in the Administrators tabbed window.

4. Click **Apply**. The new administrators are added to the enterprise domain.

To remove a user from the list of Enterprise Domain Administrators:

1. In the left navigator pane, select the enterprise domain from which you wish to remove administrators.

2. In the right pane, select the **Administrators** tab.

3. Select a user from the list of administrators.

4. Click **Remove**. The selected user is removed from the list.

5. Click **Apply**. The user is removed as an Enterprise Domain Administrator for that domain in the realm.

Managing Enterprise Domain Database Schema Mappings

Database **schema mapping**s, also referred to as user schema mappings, enable databases registered in the directory to accept connections from users without requiring any dedicated database schemas for them. For example, when local user Scott connects to a database, that logon can succeed only if a database schema called Scott exists. This requirement can be difficult to maintain if an enterprise has thousands of users and hundreds of databases.

Users defined in an LDAP-compliant directory do not require dedicated schemas on every Oracle9*i* or later database to which they might connect.

A database can use a schema mapping to share one database schema among multiple directory users. Each schema mapping is a pair of values, the base in the directory at which users exist and the name of the database schema they will use.

After selecting either a database or a domain in the main application tree, you can use the Database Schema Mappings window to manage database schema mappings. If a domain is selected, these mappings apply to all databases in the enterprise domain. However, for the mapping to be effective in a database, that database must have a schema with the name used in the mapping. This window contains a list of database schema names, directory DNs, and mapping types (Figure 4–9):

Figure 4–9   Enterprise Security Manager: Database Schema Mappings Tab
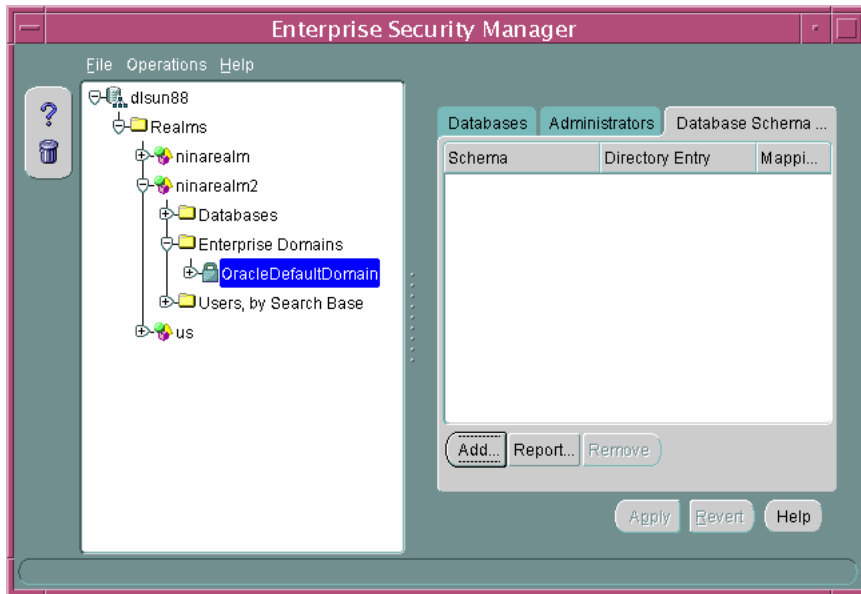
To add a new mapping to the list of database schema mappings in the enterprise domain:

1. In the Database Schema Mapping tabbed window, click **Add**.

    The Add Database Schema Mappings window appears (Figure 4–10).

    The three components in this window are the directory search tree, the options for choosing either subtree or entry-level mapping, and the field for entering the schema name. (You select the user's DN or the base of users in the directory tree.)

    In this window, you make the database schema mapping by selecting a base in the directory and pairing it with a database schema name, as the following steps illustrate.

Figure 4–10   Enterprise Security Manager: Add Database Schema Mappings Window
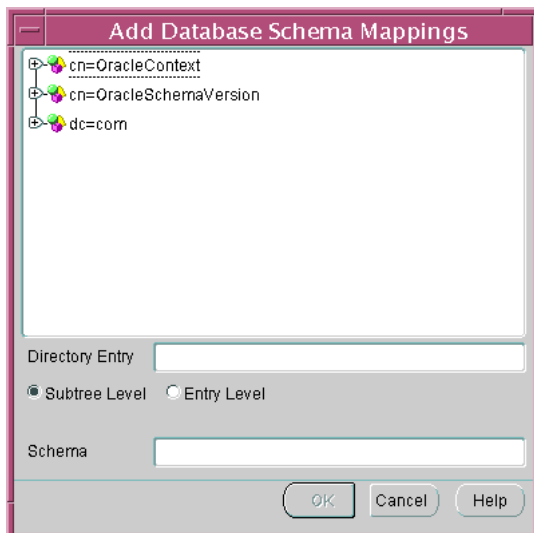
2. Navigate the directory to select a desired entry as a base for the database schema mapping. Although any directory entry can be selected, you should, for an entry-level mapping, select the actual user. For a subtree-level mapping, select the entry located above the subtree of users to be mapped. You can also edit the contents of the Directory Entry field in this window to manually define the base.

   Note that subtree-level mapping is usually the most useful.

3. Click the option corresponding to your chosen mapping type, **Subtree Level** or **Entry Level**.

4. In the Schema field, enter the name of the database schema for which this Mapping will be made, and click **OK**. This schema name must be a valid name that identifies a shared schema already existing on that database. Your new database schema mapping appears in the database schema mappings window (Figure 4–9).

5. Click **Apply**. The new database schema mapping is added to the selected database or domain in the realm.

To remove a mapping from the list of database schema mappings in an enterprise domain:

1. Select a mapping by selecting from the Database Schema Mapping tabbed window.

2. Click **Remove**. The selected mapping is removed from the list.

3. Click **Apply**. The mapping is removed from the enterprise domain.

Managing Password-Accessible Domains

The following three requirements enable a database to accept a connection from a password-authenticated user:

1. The database must be a member of a domain configured to accept password authentication (See: Table 4–4 on page 4-12).

2. The domain must be a member of a password-accessible domains group, called the **Password-Accessible Domains List**.

   (The domain can be added by a member of either of two groups, the OracleContextAdmins group or the OracleDBSecurityAdmins directory administrator group. Domain members of this list, which are databases, can read the user's password verifier in the directory. Databases excluded from this list cannot read the user's password verifier in the directory.)

3. The directory subtree in which the user entry appears must be enabled for Oracle Database access.

To configure password accessibility:

1. Select the enterprise domain in the left navigator pane.

2. Select the **Databases** tabbed window and select **Password** or **All Types** from the **User Authentication** methods listed. (See Figure 4–7 on page 4-11)

3. Click **Apply**.

To add a domain to the Password-Accessible Domains List:

1. Select the identity management realm in the left navigator pane.

2. Select the **Accessible Domains** tabbed window and click **Add**. The Add Accessible Enterprise Domains dialog box appears. See Figure 4–11.

Figure 4–11 Enterprise Security Manager: Add Accessible Enterprise Domains Dialog Box
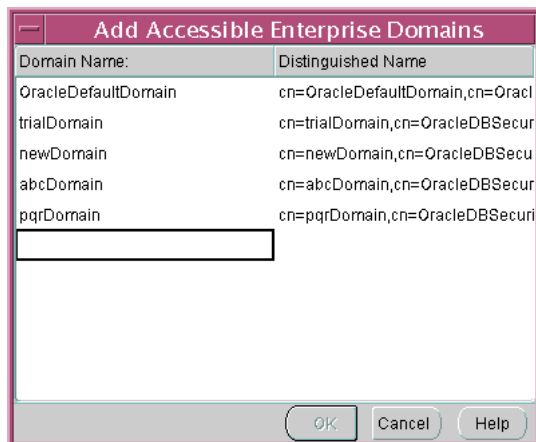
3. Select **OracleDefaultDomain** from the list of enterprise domains and click **OK**. OracleDefaultDomain is added to the list of domains in the Accessible Domains tab.

4. Click **Apply**. The new domain is added to the password accessible domains list in the directory.

> **Note:**
>
> ■ By default, the `cn=Users` subtree in an identity management realm has ACLs (access control lists) to enable appropriate database access to user password attributes. If you do not use this subtree to store users, then refer to *Oracle Internet Directory Administrator's Guide* for information about setting up proper ACLs for another user search base.
>
> ■ OracleDefaultDomain is a member of the password-accessible domains list by default, but it can be removed.

To remove an enterprise domain from the password-accessible domains list:

1. Select the identity management realm in the left navigator pane.

2. Select the **Accessible Domains** tabbed window and select the enterprise domain you want to remove from the list.

3. Click **Remove**. The domain is removed from the list under the Accessible Domains tab.

4. Click **Apply**. The domain is removed from the password-accessible domains list in the directory.

See Also:

■ "Defining Database Membership of an Enterprise Domain" on page 4-11

■ "Managing Database Security Options for an Enterprise Domain" on page 4-12

Managing Database Administrators

Database administrators are members of the directory administrative group OracleDBAdmins. They are directory users who have privileges to modify the database entry and its subtree in the realm. These privileges include creating and managing **user-schema mapping**s. When a user registers a database in the directory, Database Configuration Assistant automatically puts that user into the OracleDBAdmins group.

Database administrators can be managed by using the Administrators tabbed window when a database is selected under a realm in the navigator pane. See Figure 4–12 on page 4-17. Note that only OracleDBAdmins or **OracleContextAdmins** group members can add or remove users from the OracleDBAdmins group.

Figure 4–12   Enterprise Security Manager: Administrators Tab
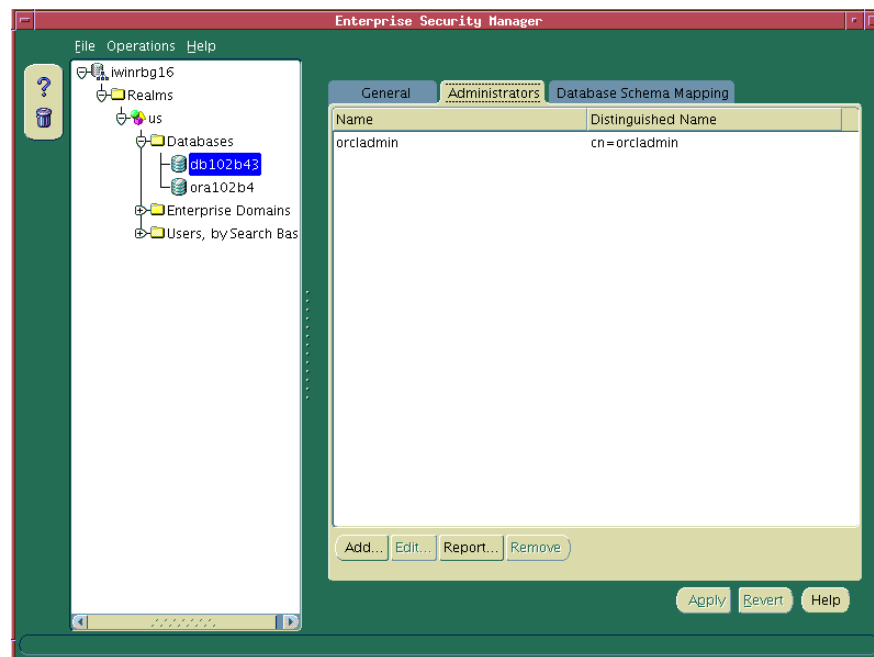
To remove a user from the list of Database Administrators:

1. In the Administrators tabbed window, select a user from the list of administrators.

2. Click **Remove**. The selected user is removed from the list.

3. Click **Apply**. The user is removed as a Database Administrator for that database.

To add a new user to the list of Database Administrators:

1. Click **Add** in the Administrators tabbed window. The Add Users window appears. Use this window to locate and select users in the directory.

2. Select a user or users from the directory to be added as a database administrator. The new users are displayed in the Administrators tabbed window.

3. Click **Apply**. The new administrators are added to the database in the realm.

> **See Also:**
>
> - "Creating New Enterprise Users" on page 4-6
> - "Browsing Users in the Directory" on page 4-8

Managing Proxy Permissions in Enterprise Domains

An enterprise domain within an identity management realm can contain multiple proxy permissions. A proxy permission object is a set of authorizations, which can be granted to enterprise users to proxy as local database users within the enterprise domain.

Selecting the Proxy Permissions node under a domain in the navigation tree shows two lists in the right main window of Enterprise Security Manager. The first list identifies the proxy permission names. The other list shows the corresponding DNs of the proxy permission entries in Oracle Internet Directory. Each proxy permission under the Proxy Permissions container includes a list of users and groups to which it applies. It also includes the list of target database users to which the proxy users are mapped.

To enable enterprise users to proxy as local database users, the following steps are required:

- Granting Enterprise Users Access to Local Database Schemas
- Listing Specific Enterprise Users Who Will Proxy
- Linking Those Enterprise Users to the Target Database Schemas

Granting Enterprise Users Access to Local Database Schemas

To allow enterprise users the right to proxy as local database users, an administrator must run the following command for each such local database user:

```
ALTER USER local_database_user_name GRANT CONNECT THROUGH ENTERPRISE USERS
```

Listing Specific Enterprise Users Who Will Proxy

This task requires the following steps to be performed in Enterprise Security Manager:

1. In the navigation tree, select the realm containing the domains in which you want to establish enterprise user proxies.

2. Select the domain containing the local databases to which users will proxy. From the **Operations** menu, choose **Create Proxy Permission**. The dialog box shown in Figure 4–13 appears.

Figure 4–13   Create Proxy Permission Dialog Box
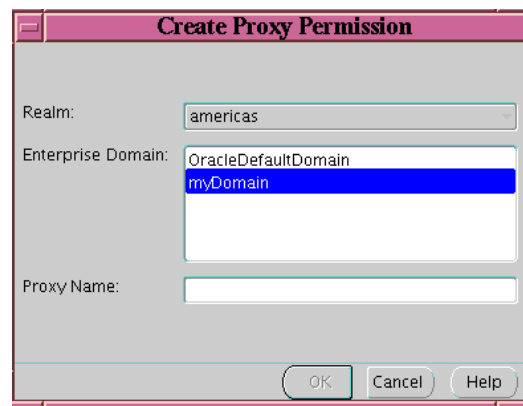
3. Type a name for the proxy permission in the Proxy Name text field. Click **OK**.

4. Figure 4–14 illustrates how the newly created proxy permission entry appears in the navigation tree:

Figure 4–14   Tree View of New Proxies in Proxy Permissions
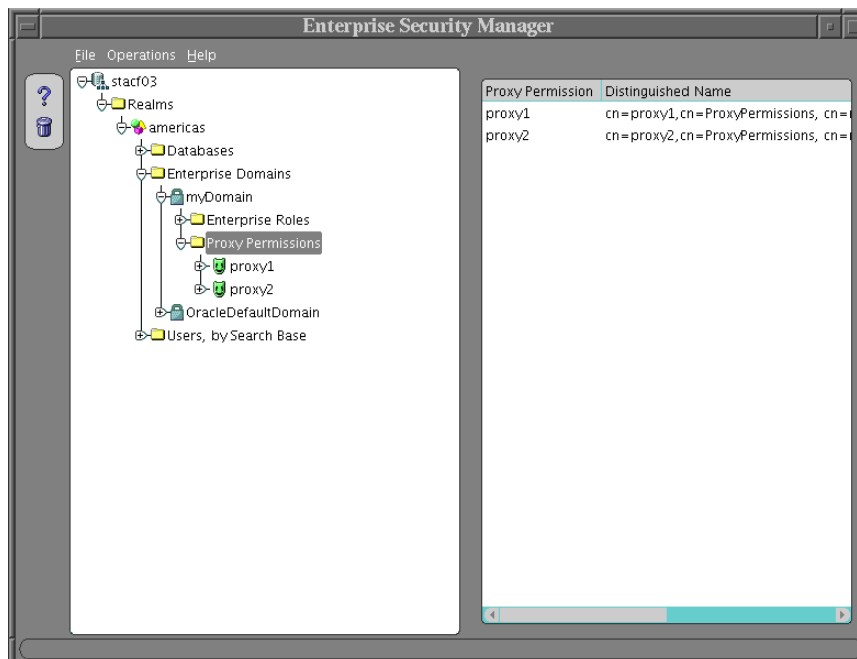
5. Select a proxy. The associated tabbed windows appear: one named **Proxy Users and Groups**, the other named **Database Target Users**, as shown in Figure 4–15.

Figure 4–15   Proxy Selected in Proxy Permissions
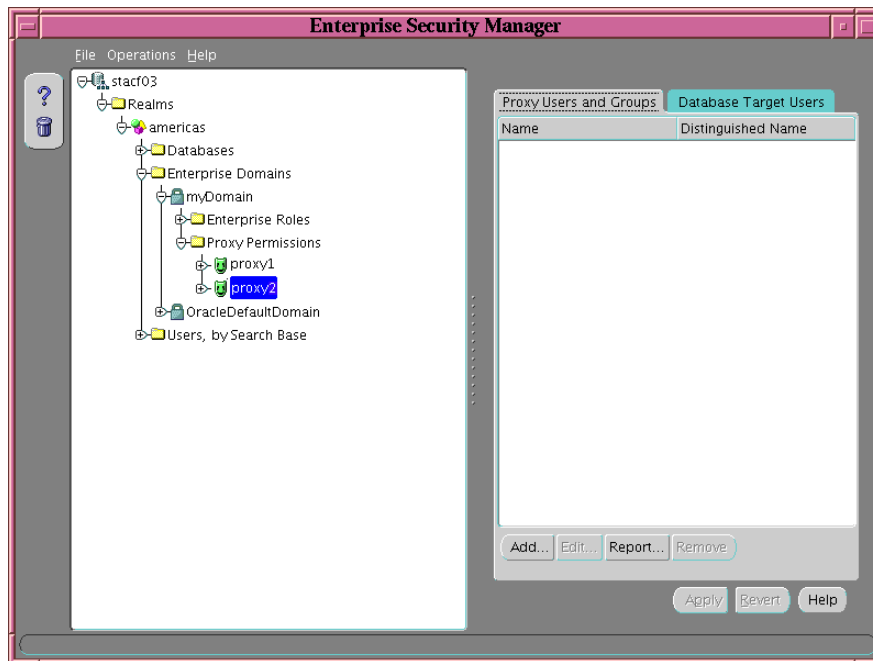
6. Select the **Proxy Users and Groups** window and click **Add.** The **Add Enterprise Users** window appears.

7. Use the Add Enterprise Users window to select one or more enterprise users to be granted proxy permissions. Your window will look like Figure 4–16:

Figure 4–16   Selecting an Enterprise User
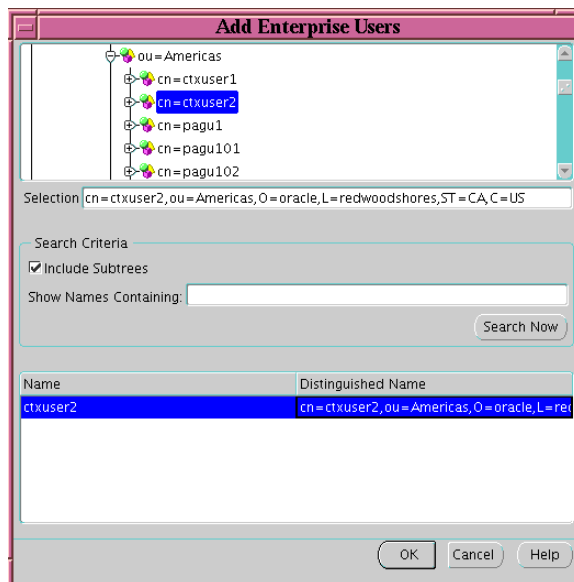
8. Click **OK**. Your selected enterprise user now appears as an enterprise user for proxy2, with his or her name and Distinguished Name in the Proxy Users and Groups tabbed window, as shown in Figure 4–17:

Figure 4–17   Enterprise User with Proxy Permission
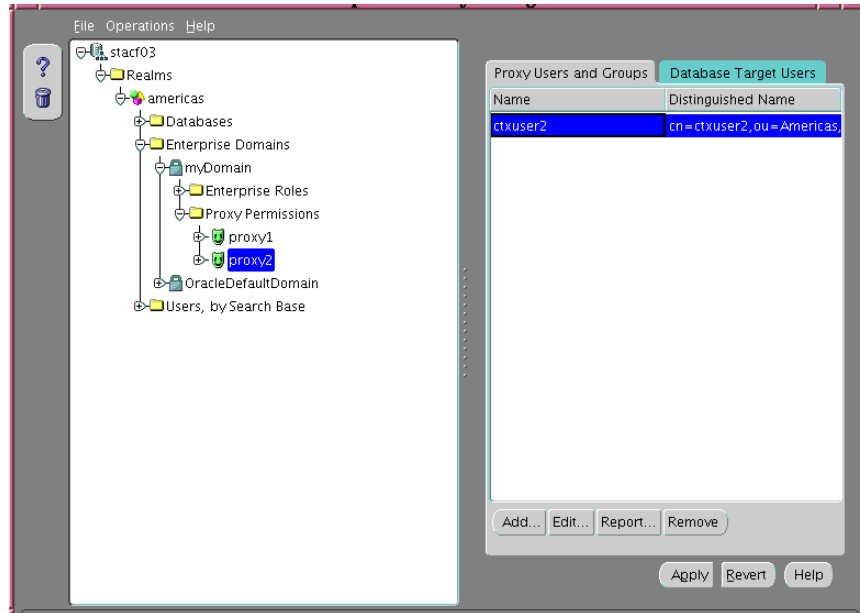
9.   Click **Apply**.

Linking Those Enterprise Users to the Target Database Schemas

The preceding section identified the enterprise users who will be permitted to proxy as local database users. In this section, you select the target databases and local users to whom those enterprise users can proxy.

1.   Select the **Database Target Users** tab and click **Add**.

2.   The **Add Target Users** window appears as shown in Figure 4–18. It shows the available databases in the domain.

Figure 4–18   Databases in the Domain
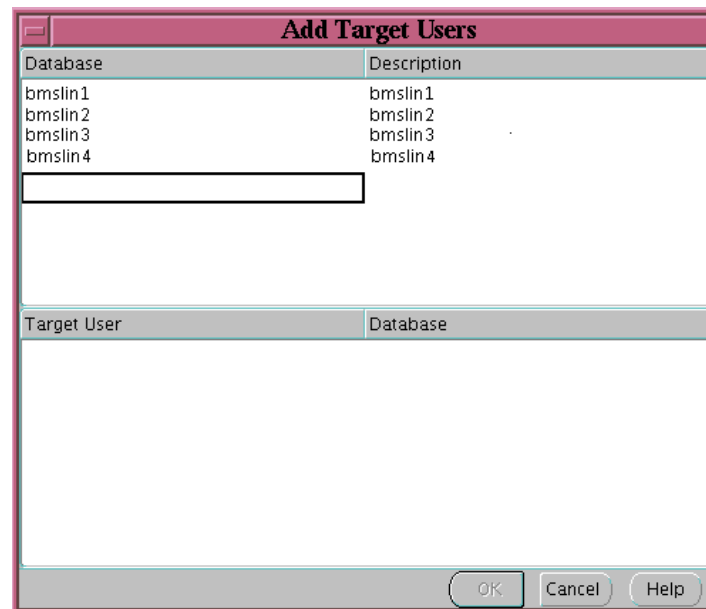
3. Select the desired database from which the target users will be chosen. The login screen for that database will appear, as illustrated in Figure 4–19.

Figure 4–19   Database Login Screen
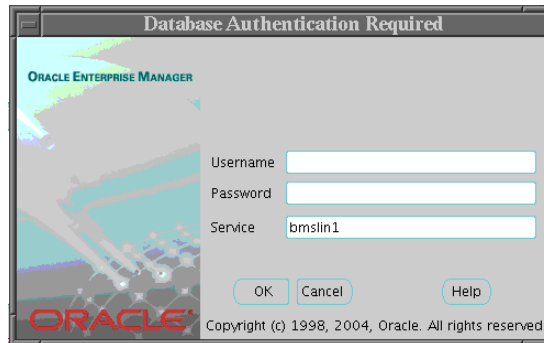
4. After you log in, the available users will be listed as illustrated in Figure 4–20. (This list will include only those users for whom an earlier ALTER USER command was issued, as described in "Granting Enterprise Users Access to Local Database Schemas".)

Figure 4–20   Local Database Users Available as Proxy Targets
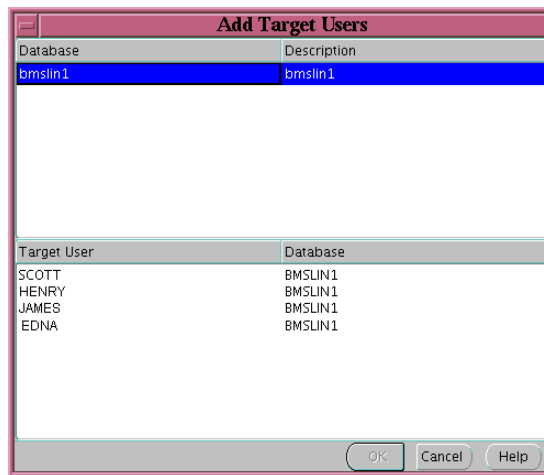
5. Select the users you want, and click **OK**. A screen similar to the one shown in Figure 4–21 appears.

Figure 4–21   Users Selected as Proxy Targets
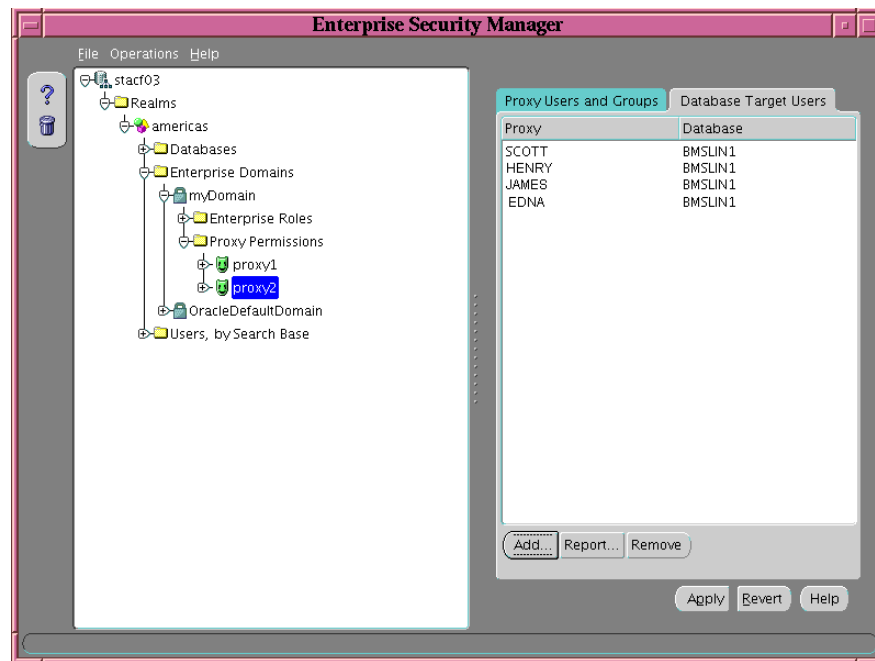
6. When you click **Apply**, these local database users will be established as legitimate proxy targets for your selected enterprise users.

Roles: Administering 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.

This section describes how to use Enterprise Security Manager to administer enterprise roles in the directory. It contains the following topics:

- Creating a New Enterprise Role
- Assigning Database Global Role Membership to an Enterprise Role
- Granting Enterprise Roles to Users

Creating a New Enterprise Role

You can create an enterprise role in an enterprise domain either from the Operations menu on the Enterprise Security Manager main window (Figure 4–9), or by right-clicking an enterprise domain in the main application tree. In either case, the Create Enterprise Role window appears (Figure 4–22):

Figure 4–22   Enterprise Security Manager: Create Enterprise Role Window
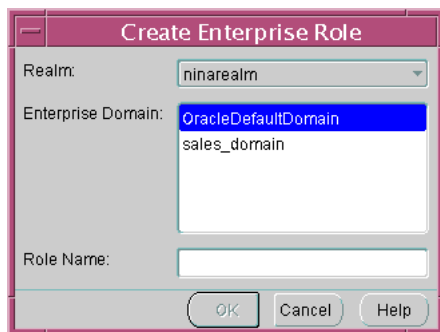
To create a new enterprise role:

1. Select the identity management realm from the list. This realm contains the enterprise domain to hold the new enterprise role.

    > **Note:**   If you invoked the Create Enterprise Role window by right-clicking an enterprise domain, then the name of the identity management realm is already selected.

2. From the Enterprise Domain list, select the appropriate enterprise domain for the new enterprise role.

    > **Note:**   If you invoked the Create Enterprise Role window by right-clicking an enterprise domain, then the name of the enterprise domain is already selected.

3. Enter the name of the new enterprise role in the **Role Name** field.

4. Click **OK**. The new enterprise role is created in the enterprise domain and appears on the main application tree.

To remove an enterprise role:

1. Select the enterprise role from the main application tree (Figure 4–9).

2. Select **Remove Enterprise Role**, either from the Operations menu or by right-clicking the enterprise domain in the main application tree.

3. Enterprise Security Manager prompts you to confirm the removal of the enterprise role. Click **Yes**.

Assigning Database Global Role Membership to an Enterprise Role

Use the Database Global Roles tabbed window (Figure 4–23) of the Enterprise Security Manager main window to manage database global role membership in an enterprise role. This window lists the names of each **global role** that belongs to the enterprise role, along with the name of the database on which that global role exists.

Figure 4–23   Enterprise Security Manager: Database Global Roles Tab
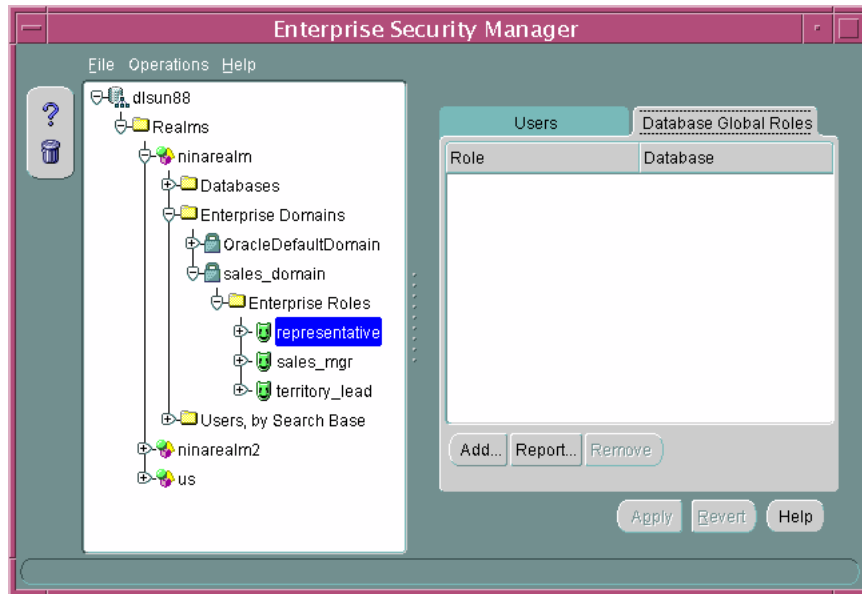
When populating an enterprise role with different database roles, you can only use roles configured to be *global roles* on the databases you select. A global role on a database is identical to a normal role, except that the **database administrator** has defined it to be authorized only through the directory (Global roles are created with the syntax, `CREATE ROLE <role_name> IDENTIFIED GLOBALLY;`). A database administrator cannot locally grant and revoke global roles to users of the database.

To add a global role to an enterprise role:

1. Click **Add** (Figure 4–23). The Add Global Database Roles window appears. This window lists all of the databases in the enterprise domain from which global roles can be selected to add to an enterprise role.

2. Select a database from which to obtain global roles. A window appears and prompts you for logon details to authenticate to the database (and fetch global roles). Typically, a DBA logon to that database is required.

    Note that the name of the database appears in the Service field by default. You can use this name to connect to the database in two circumstances:

    ■   If your Oracle home has LDAP enabled as its Oracle Net naming method, or

    ■   If this name appears as a TNS alias in your local Oracle Net configuration

    Otherwise, you can overwrite the content of the Service field with

    ■   any other TNS alias configured for that database, or

    ■   a connect string in the format `<host>:<port>:<oracle sid>`. For example, `cartman:1521:broncos`

Figure 4–24   Enterprise Security Manager: Database Authentication Required Window

1. Click **OK**. Enterprise Security Manager connects you to the given database and fetches the list of global roles supported on that database. The list of values, if any, is displayed in the Add Global Database Roles window.

2. Select one or more global roles from the list of returned values and click **OK**. These global roles appear in the Database Global Roles tabbed window (Figure 4–23).

3. Click **Apply**. The new global roles are added to the enterprise role in the enterprise domain.

To remove a database global role from an enterprise role:

1. Select a global role under the Database Global Roles tab, and click **Remove**. The global role is removed from the list.

2. Click **Apply**. The global role is removed from the enterprise role in the enterprise domain.

Granting Enterprise Roles to Users

You can grant an enterprise role to users in two ways:

- you can select a user and add a role (see "Defining an Initial Enterprise Role Assignment" on page 4-7), or

- you can select a role and add a user

An enterprise role granted to a user includes all database global roles contained within that enterprise role. Use the Users tabbed window.

To grant an enterprise role to users:

1. Select the role in the navigation tree and click **Add** in the Users tabbed window. The **Add Enterprise Users** window appears. Use this window to locate and select one or more directory users to add as enterprise role grantees (Figure 4–25):

Figure 4–25   Enterprise Security Manager: Add Enterprise Users Window
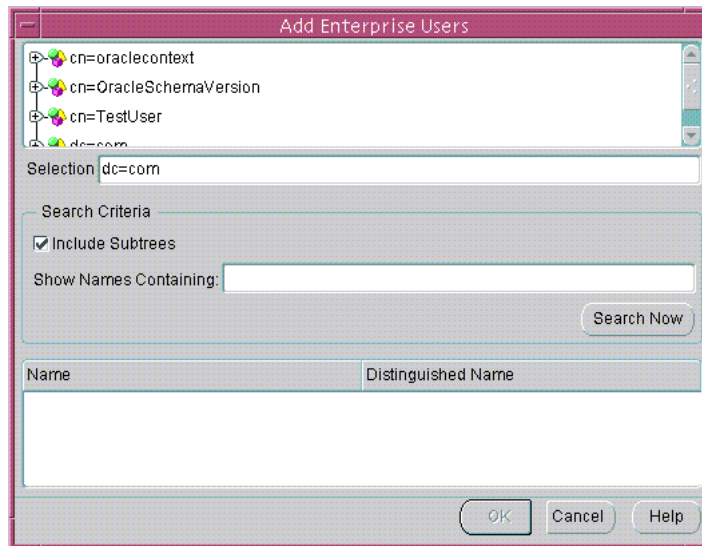
2. Select a user or users and click **OK**. The new grantees are added to the list of users who have that enterprise role in the enterprise domain.

3. Click **Apply**. The user or users are granted the selected enterprise role.

To remove a user from the list of enterprise role grantees:

1. Select a user from the list of grantees in the Users tabbed window.

2. Click **Remove**. The selected user is removed from the list.

3. Click **Apply**. The user is removed as a grantee for that enterprise role in the enterprise domain.

A

Using the User Migration Utility

This chapter describes the User Migration Utility. You can use it to perform bulk migrations of database users to an LDAP directory, where they can be stored and managed centrally as enterprise users. It contains the following topics:

- Benefits of Migrating Local or External Users to Enterprise Users
- Introduction to the User Migration Utility
- Prerequisites for Performing Migration
- User Migration Utility Command-Line Syntax
- Accessing Help for the User Migration Utility
- User Migration Utility Parameters
- User Migration Utility Usage Examples
- Troubleshooting Using the User Migration Utility

Benefits of Migrating Local or External Users to Enterprise Users

Migrating from a database user model to an enterprise user model provides solutions to administrative, security, and usability challenges in an enterprise environment. In an enterprise user model, all user information is moved to an LDAP directory service.

Enterprise user security provides the ability to easily and securely manage enterprise wide users by providing the following benefits:

- Centralized storage of user credentials, roles, and privileges in an LDAP version 3-compliant directory server
- Provides the infrastructure to enable single sign-on using X.509v3-compliant certificates, typically deployed where end-to-end SSL is required
- Enhanced security through more timely maintenance and fewer user passwords

  The centralization of user information inherent in the enterprise user model makes it easier to manage. Security administrators can perform necessary maintenance changes to user information immediately, thereby maintaining better control over access to critical network resources. In addition, users find the enterprise user model easier to use because they have fewer passwords to remember. So, they are less likely to choose easily guessed passwords or to write them down where others can copy them.

> **See Also:** "Introduction to Enterprise User Security" on page 1-1 for detailed conceptual information about enterprise user security

Introduction to the User Migration Utility

The User Migration Utility is a command-line utility that enables enterprise user administrators to move their users from a local database model to an enterprise user model. You can easily migrate thousands of local and external database users to an enterprise user environment in an LDAP directory where they can be managed from a central location. The utility connects to the database using the Oracle JDBC OCI driver.

Enterprise user administrators can select for migration any combination of the following user subsets in a database:

- List of users specified on the command line or in a file

- All external users

- All global users

In addition, enterprise user administrators can specify values for utility parameters that determine how the users are migrated such as

- Where to put the migrated users in the LDAP directory tree

- Map a user with multiple accounts on various databases to a single directory user entry

The following sections explain the migration process and the changes that occur to user schemas.

Note: After external users are migrated, their external authentication and authorization mechanisms are replaced by directory-based mechanisms. New passwords are randomly generated for migrated users if they are mapped to newly created directory entries.

Bulk User Migration Process Overview

Bulk user migration is a two-phase process. In Phase One, you start the migration process by populating user information into an interface database table. Enterprise user administrators then verify that the information is accurate before completing the migration with Phase Two, which commits the changes to the database and the directory. The process is described in the following steps:

- Step 1: (Phase One) Preparing for the Migration

- Step 2: Verify User Information

- Step 3: (Phase Two) Completing the Migration

Step 1: (Phase One) Preparing for the Migration

In the first part of the migration process, the utility checks if the `ORCL_GLOBAL_USR_ MIGRATION_DATA` interface table exists in the enterprise user administrator's schema. If it exists, then the administrator can choose to reuse the table (clearing its contents), reuse the table and its contents, or re-create the table. Phase One can be run multiple times, each time adding to the interface table. If the table does not exist, then the utility creates it in the administrator's schema. The interface table is populated with information about the migrating users from the database and the directory. The command-line options used determine what information populates this table.

> **Note:** The utility will not create the interface table in the `SYS` schema.

Step 2: Verify User Information

This is an intermediate step to allow the enterprise user administrator to verify that the user information is correct in the interface table before committing the changes to the database and the directory.

Step 3: (Phase Two) Completing the Migration

After the interface table user information is checked, Phase Two begins. The utility retrieves the information from the table and updates the directory and the database.

Depending on whether directory entries exist for migrating users, the utility creates random passwords as follows:

- If migrating users are being mapped to newly created directory entries, then the utility generates random passwords, which are used as credentials for both the database and directory.

- If migrating users are being mapped to existing directory entries with unset database passwords, then the utility generates random database passwords only.

In either case, after generating the required random passwords, the utility then stores them in the `DBPASSWORD` and `DIRPASSWORD` interface table columns. The enterprise user administrator can read these passwords from the interface table and inform migrating users.

> **See Also:** "User Migration Utility Parameters" on page A-8 for a list of command-line options and their descriptions

About the ORCL_GLOBAL_USR_MIGRATION_DATA Table

This is the interface table which is populated with information about the migrating users during Phase One of the bulk user migration process. The information that populates this table is pulled from the database and checked against existing entries in the directory. If there is corresponding information in the directory, then that is marked in the table for that user. After enterprise user administrators verify the information in this table, changes are made to the directory and the database in Phase Two.

> **Caution:** The `ORCL_GLOBAL_USR_MIGRATION_DATA` interface table contains very sensitive information. Access to it should be tightly controlled using database privileges.

The table columns are listed in Table A–1.

Table A–1   ORCL_GLOBAL_USR_MIGRATION_DATA Table Schema

Column Name	Data Type	Null	Description
`USERNAME` (Primary Key)	VARCHAR2(30)	NOT NULL	Database user name
`OLD_SCHEMA_TYPE`	VARCHAR2(10)	-	Old schema type in the database before migration
`PASSWORD_VERIFIER`	VARCHAR2(30)	-	Not used

Table A–1   (Cont.) ORCL_GLOBAL_USR_MIGRATION_DATA Table Schema

Column Name	Data Type	Null	Description
USERDN	VARCHAR2(4000)	-	Distinguished Name (DN) of the user in the directory (new or existing)
USERDN_EXIST_FLAG	CHAR(1)	-	Flag indicating whether the DN already exists in the directory
SHARED_SCHEMA	VARCHAR2(30)	-	Shared schema name, if users are to be mapped to a shared schema during phase two
MAPPING_TYPE	VARCHAR2(10)	-	Mapping type (database or domain)
MAPPING_LEVEL	VARCHAR2(10)	-	Mapping level (entry or subtree)
CASCADE_FLAG	CHAR(1)	-	Cascade flag used when dropping a user (for shared schema mapping only)
DBPASSWORD_EXIST_FLAG	CHAR(1)	-	Flag indicating whether the database password verifier already exists in the directory for this user
DBPASSWORD	VARCHAR2(30)	-	Randomly generated database password verifiers to be stored in the directory
DIRPASSWORD	VARCHAR2(30)	-	Randomly generated directory password for new entries
PHASE_COMPLETED	VARCHAR2(10)	-	Information about the phase that has been completed successfully
NEEDS_ATTENTION_FLAG	CHAR(1)	-	Flag indicating whether the row contains abnormalities that require administrator attention
ATTENTION_DESCRIPTION	VARCHAR2(100)	-	Textual hint for the administrator if the attention flag is set
KERBEROS_PNAME	VARCHAR2(30)	-	Kerberos Principal Name for external kerberos users

Which Interface Table Column Values Can Be Modified Between Phase One and Phase Two?

After running phase one of the utility, if necessary, enterprise user administrators can change the interface table columns listed in Table A–2.

Table A–2   Interface Table Column Values That Can Be Modified Between Phase One and Phase Two

Column Name	Valid Values	Restrictions
USERDN	DN of user	If this value is changed, then the administrator should verify that the USERDN_EXIST_FLAG and the DBPASSWORD_EXIST_FLAG values are set accordingly.
USERDN_EXIST_FLAG	T/F	If the USERDN column value changes, then this column value should also change to reflect the new USERDN status.
DBPASSWORD_EXIST_FLAG	T/F	If the USERDN column value changes, then this column value should also change to reflect whether a database password exists for the new USERDN.
SHARED_SCHEMA	Shared schema name	Specify only if a shared schema exists in the database.
MAPPING_TYPE	DB/DOMAIN	Set this value only if SHARED_SCHEMA is not set to NULL.

Table A–2   (Cont.) Interface Table Column Values That Can Be Modified Between Phase One and Phase

Column Name	Valid Values	Restrictions
MAPPING_LEVEL	ENTRY/SUBTREE	Set this value only if SHARED_SCHEMA is not set to NULL.
CASCADE_FLAG	T/F	Set this value only if SHARED_SCHEMA is not set to NULL. If this column is set to true (T), then the users' schema objects are forcibly deleted. If this column is set to false (F), then the administrator must delete all user schema objects before going into Phase Two.
PHASE_COMPLETED	ZERO/ONE/TWO	If the administrator can resolve the conflicts or ambiguities specified with the NEEDS_ATTENTION_FLAG, then this column value can be changed to ONE so phase two can be run with the utility.

Migration Effects on Users' Old Database Schemas

If shared schema mapping is not used, then users retain their old database schemas. If shared schema mapping is used, then users' local schemas are dropped from the database, and they are mapped to a shared schema that the enterprise user administrator creates for this purpose before performing the migration. When migrated users own database objects in their old local database schemas, administrators can specify that the schema and objects are not to be dropped by setting the CASCADE parameter to NO. When the CASCADE parameter is set to NO, users who own database objects in their old local schemas do not migrate successfully so their objects are not dropped.

If some users want to retain the objects in their local database schemas and be mapped to a shared schema, then the administrator can manually migrate those objects to the shared schema before performing the bulk user migration. However, when objects are migrated to a shared schema, they are shared among all users who share that new schema.

Table A–3 summarizes the effects of setting the MAPSCHEMA and CASCADE parameters.

Table A–3   Effects of Choosing Shared Schema Mapping with CASCADE Options

MAPSCHEMA Parameter Setting	CASCADE Parameter Setting	User Migration Successful?	User Schema Objects Dropped?
PRIVATE	NO (default setting)	Yes	No
SHARED	NO	Yes[1]	No
SHARED	YES	Yes[2]	Yes

[1]   Users migrate successfully only if they do not own objects in their old database schemas; otherwise, they fail.
[2]   Users migrate successfully, and their old database schemas are dropped.

> **See Also:**   "User Migration Utility Parameters" on page A-8 for detailed information about the MAPSCHEMA, CASCADE, and other parameters that can be used with this utility

Migration Process

Enterprise users are defined and managed in the directory and can be authenticated to the database either with a password or with a certificate. Users who authenticate with a password require an Oracle Database password, which is stored in the directory. Users who authenticate with a certificate must have a valid X.509 v3 certificate.

This utility performs the following steps during migration:

1. Selects the users from the database for migration.

2. Creates corresponding user entries or uses existing entries in the directory.

3. Creates new database passwords and copies the corresponding verifiers to the directory for migrating users.

4. Puts the schema mapping information for the migrating users' entries in the directory. (optional)

5. Drops or alters the migrating users' local database schemas. (optional)

> **Note:** In the current release, the utility migrates users with certificate-based authentication and makes them ready for password authentication. Previously, SSL-based authenticated users were required to reset their Oracle Database passwords. User wallets are not created as part of this process.

> **See Also:** The chapter about Oracle Wallet Manager in *Oracle Database Advanced Security Administrator's Guide* for information about creating, managing, and using Oracle wallets

Prerequisites for Performing Migration

The User Migration Utility is automatically installed in the following location when you install Oracle Database Client:

```
$ORACLE_HOME/rdbms/bin/umu
```

The following sections describe what programs must be running and what user privileges are required to successfully migrate users with the User Migration Utility.

Required Database Privileges

To successfully use this utility, enterprise user administrators must have the following database privileges:

- ALTER USER

- DROP USER

- CREATE TABLE

- SELECT_CATALOG_ROLE

These privileges enable the enterprise user administrator to alter users, drop users, look at dictionary views, and create the interface table that is used by this utility.

Required Directory Privileges

In addition to the required database privileges, enterprise user administrators must have the directory privileges which allow them to perform the following tasks:

- Create entries in the directory under the specified user base and Oracle context location

- Browse the user entries under the search bases

Required Setup to Run the User Migration Utility

Perform the following steps before using the User Migration Utility:

1. Ensure that the directory server is running with SSL enabled for no authentication.

2. Ensure that the database server is running with encryption and integrity enabled.

3. Ensure that the database listener has a TCP listening end point.

4. Create an identity management realm in the directory, if it does not already exist.

5. Create the parent context for the user entries in the directory, if it does not already exist. The default (and recommended) location is in the `orclcommonusercreatebase` subtree in the common container in the Oracle Context.

6. Set up directory access for the database Oracle home by using Oracle Net Configuration Assistant to create an `ldap.ora` file. Note that the `ldap.ora` file must include the identity management realm DN so the utility can locate the correct administrative context. The utility searches for this file under `$LDAP_ADMIN`, `$ORACLE_HOME/ldap/admin`, `$TNS_ADMIN`, `$ORACLE_HOME/network/admin`, and, finally, the Domain Name System (DNS) server, if you are using DNS discovery. (See *Oracle Internet Directory Administrator's Guide* for information about DNS server discovery.)

> **Note:**
>
> - If you plan to use shared schema mapping when migrating users, then you must create the shared schema before running this utility.
>
> - The same `ldap.ora` file must be used for both Phase One and Phase Two of a user migration.

> **See Also:**
>
> - Chapter 3, "Enterprise User Security Configuration Tasks and Troubleshooting" for detailed information about setting up enterprise user authentication after the user migration is finished
>
> - *Oracle Internet Directory Administrator's Guide*

User Migration Utility Command-Line Syntax

To perform a bulk migration of database users to enterprise users, use the following syntax:

```
umu parameter1 parameter2 ...
```

For parameters that take a single value use the following syntax:

```
keyword=value
```

For parameters that take multiple values, use a colon (:) to separate the values as in the following syntax:

```
keyword=value1:value2:...
```

Example A–1 shows the syntax used to run the utility through both phases of the bulk user migration process.

Example A–1   User Migration Utility Command-Line Syntax

```
umu PHASE=ONE
DBADMIN=dba_username:password
ENTADMIN=enterprise_admin_DN:password
USERS=[ALL_GLOBAL | ALL_EXTERNAL | LIST | FILE]
DBLOCATION=database_host:database_port:database_sid
DIRLOCATION=ldap_directory_host:ldap_directory_port
USERSLIST=username1:username2:username3:...
USERSFILE=filename
MAPSCHEMA=[PRIVATE | SHARED]:schema_name
MAPTYPE=[DB | DOMAIN]:[ENTRY | SUBTREE]
CASCADE=[YES | NO]
CONTEXT=user_entries_parent_location
LOGFILE=filename
PARFILE=filename
KREALM=ACME.COM

umu PHASE=TWO
DBADMIN=dba_username:password
ENTADMIN=enterprise_admin_DN:password
DBLOCATION=database_host:database_port:database_sid
DIRLOCATION=ldap_directory_host:ldap_directory_port
LOGFILE=filename
PARFILE=filename
```

> **Note:**   If the enterprise user administrator does not specify the mandatory parameters on the command line, then the utility will prompt the user for those parameters interactively.

See Also:

- "User Migration Utility Parameters" on page A-8 for a complete list of all available parameters and detailed information about them

- "User Migration Utility Usage Examples" on page A-15 for examples of typical utility uses

Accessing Help for the User Migration Utility

To display the command-line syntax for using the User Migration Utility, enter the following command at the system prompt:

```
umu HELP=YES
```

While the HELP parameter is set to YES, the utility cannot run.

User Migration Utility Parameters

The following sections list the available parameter keywords and the values that can be used with them when running this utility. The keywords are not case-sensitive.

Keyword: HELP

Attribute	Description
Valid Values:	YES or NO (These values are not case-sensitive.)
Default Setting:	NO
Syntax Examples:	HELP=YES
Description:	This keyword is used to display Help for the utility. YES displays the complete command-line syntax. To run a command, set the value to NO, or do not specify a value for the parameter to accept the default.
Restrictions:	None

Keyword: PHASE

Attribute	Description
Valid Values:	ONE or TWO (These values are not case-sensitive.)
Default Setting:	ONE
Syntax Examples:	PHASE=ONE
	PHASE=TWO
Description:	Indicates the phase for the utility. If it is ONE, then the utility populates the interface table with the information specified in the command-line arguments and the existing user entries in the directory. If it is TWO, then the utility uses the information that is available in the interface table and updates the directory and the database.
Restrictions:	None

Keyword: DBLOCATION

Attribute	Description
Valid Values:	*host:port:sid*
Default Setting:	No default setting
Syntax Examples:	DBLOCATION=my_oracle.us.oracle.com:7777:ora902
Description:	Provides the host name, port number, and SID for the database instance
Restrictions:	■ This parameter is mandatory.
	■ The value for this parameter must be the same for both Phase One and Phase Two.
	■ The database should be configured for encryption and integrity.

Keyword: DIRLOCATION

Attribute	Description
Valid Values:	*host:port*
Default Setting:	This value is automatically populated from the ldap.ora file by default.

Attribute	Description
Syntax Examples:	DIRLOCATION=my_oracle.us.oracle.com:636
Description:	Provides the host name and port number for the directory server where the LDAP server is running on SSL with no authentication
Restrictions:	The value for this parameter must be the same for both Phase One and Phase Two.

Keyword: DBADMIN

Attribute	Description
Valid Values:	*username:password*
Default Setting:	No default setting
Syntax Examples:	DBADMIN=system:manager
Description:	User name and password for the database administrator with the required privileges for connecting to the database
Restrictions:	■ This parameter is mandatory. ■ The username value for this parameter must be the same for both Phase One and Phase Two.

Keyword: ENTADMIN

Attribute	Description
Valid Values:	*userDN:password*
Default Setting:	No default setting
Syntax Examples:	ENTADMIN=cn=janeadmin,dc=acme,dc=com:welcome
Description:	User Distinguished Name (UserDN) and the directory password for the enterprise directory administrator with the required privileges for logging in to the directory. UserDN can also be specified within double quotation marks ("").
Restrictions:	This parameter is mandatory.

Keyword: USERS

Attribute	Description
Valid Values:	*value1:value2...*
	Values can be:
	■ `ALL_EXTERNAL` to select all external users, including those who use Kerberos and RADIUS authentication
	■ `ALL_GLOBAL` to select all global users
	■ `LIST` to specify users on the command line with "Keyword: USERSLIST"
	■ `USERSFILE` for selecting users from the file that is specified with the "Keyword: USERSFILE"
	This parameter takes multiple values. Separate values with a colon (:).
	(These values are not case-sensitive.)
Default Setting:	No default setting
Syntax Examples:	■ `USERS=ALL_EXTERNAL:ALL_GLOBAL`
	This usage instructs the utility to migrate all external users and all global users.
	■ `USERS=ALL_EXTERNAL:FILE`
	This usage instructs the utility to migrate all external users and all users specified in `USERSFILE`.
Description:	Specifies which users are to be migrated. If multiple values are specified for this parameter, then the utility uses the union of these sets of users.
Restrictions:	This parameter is mandatory for Phase One only, and it is ignored in Phase Two.

Keyword: USERSLIST

Attribute	Definition
Valid Values:	*user1:user2:...*
	Separate user names with a colon (:).
Default Setting:	No default setting
Syntax Examples:	`USERSLIST=jdoe:tchin:adesai`
Description:	Specifies a list of database users for migration. The users in this list are migrated with other users specified with the `USERS` parameter.
Restrictions:	This optional parameter is effective only when `LIST` is specified with the `USERS` parameter.

Keyword: USERSFILE

Attribute	Definition
Valid Values:	File name and path
Default Setting:	No default setting
Syntax Examples:	`USERSFILE=/home/orahome/userslist/hr_users.txt`

Attribute	Definition
Description:	Specifies a file that contains a list of database users (one user listed for each line) for migration. The users in this file are migrated with other users specified with the USERS parameter.
Restrictions:	This optional parameter is effective only when FILE is specified with the USERS parameter.

Keyword: KREALM

Attribute	Description
Valid Values:	*kerberos realm*
Default Setting:	No default setting
Syntax Examples:	KREALM=ACME.COM
Description:	Kerberos REALM for external kerberos users, which will usually be the domain name of the database server.
	If this parameter is not specified, then all external users who are considered for migration are assumed to be non-Kerberos.
Restrictions:	■ This parameter is valid only for Phase One.

Keyword: MAPSCHEMA

Attribute	Description
Valid Values:	*schema_type:schema_name*
	Schema type can be:
	■ PRIVATE
	Retains users' old local schemas. Schema name is ignored when schema type is PRIVATE. No mapping entries are created in the directory.
	■ SHARED
	Maps users to a shared schema. Mapping entries are created in the directory. Schema name specifies the shared schema name. During shared schema mapping, whether users' local schemas are dropped from the database is determined by the "Keyword: CASCADE" setting.
	(These values are not case-sensitive.)
Default Setting:	PRIVATE
Syntax Examples:	MAPSCHEMA=SHARED:HR_ALL
Description:	Specifies whether the utility populates the interface table with schema mapping information.
Restrictions:	■ See the SHARED option under Valid Values.
	■ This parameter is valid only for Phase One.

Keyword: MAPTYPE

Attribute	Description
Valid Values:	*mapping_type:mapping_level*
	Mapping type can be:
	■ DB
	■ DOMAIN
	Mapping level can be:
	■ ENTRY
	■ SUBTREE
	Separate mapping type from mapping level with a colon (:).
	(These values are not case-sensitive.)
Default Setting:	DB:ENTRY
Syntax Examples:	MAPTYPE=DOMAIN:SUBTREE
Description:	Specifies the type of schema mapping that is to be applied when "Keyword: MAPSCHEMA" is set to SHARED. If DB is specified as the mapping type, then the utility creates a mapping in the directory for the database. If DOMAIN is specified as the mapping type, then the utility creates a mapping in the directory for the domain containing the database. For domain mapping, the utility determines the domain that contains the database by an LDAP search in the relevant Oracle context.
Restrictions:	This parameter is effective only when MAPSCHEMA is set to SHARED.

> **See Also:** "About Using the SUBTREE Mapping Level Option" on page A-18 for more information about using this mapping level option

Keyword: CASCADE

Attribute	Description
Valid Values:	■ NO
	When users are mapped to a shared schema, the utility tries to drop their local schemas from the database. If this parameter is set to NO, then users are migrated only if they do not own objects in their local schema. Users who own objects in their old local schemas do not migrate. An error message is logged in the migration log file for such users.
	■ YES
	If this parameter is set to YES, then all users' schema objects are dropped along with their local schemas when they are migrated. Privileges and roles that were previously granted to the users are also revoked.
	(These values are not case-sensitive.)
Default Setting:	NO
Syntax Examples:	CASCADE=YES
Description:	Specifies whether a user's local schema is dropped when the user is mapped to a shared schema

Attribute	Description
Restrictions:	This parameter is effective only when MAPSCHEMA is set to SHARED.

Keyword: CONTEXT

Attribute	Description
Valid Values:	Distinguished Name (DN) of the parent for user entries. This is the same as the user search base or user create base in an Oracle Internet Directory identity management realm.
	Parent DN can also be specified within double quotation marks ("").
Default Setting:	Value set in *orclCommonUserCreateBase* attribute under cn=Common of Oracle Context
	Refer to Figure 1–3, "Related Entries in a Realm Oracle Context" on page 1-12 for a directory information tree diagram that shows an Oracle Context.
Syntax Examples:	CONTEXT="c=Users, c=us"
Description:	Specifies the DN of the parent entry under which user entries are created in the directory if there is no directory entry that matches the userID for the user
Restrictions:	This parameter is valid only for phase one.

Keyword: LOGFILE

Attribute	Description
Valid Values:	File name and path
Default Setting:	*$ORACLE_HOME*/network/log/umu.log
Syntax Examples:	LOGFILE=home/orahome/network/log/filename.log
Description:	Specifies the log file where details about the migration for each user are written
Restrictions:	None

Keyword: PARFILE

Attribute	Description
Valid Values:	File name and path
Default Setting:	No default setting
Syntax Examples:	PARFILE=home/orahome/network/usr/par.txt
Description:	Specifies a text file containing a list of parameters intended for use in a user migration. Each parameter must be listed on a separate line in the file. If a parameter is specified both in the parameter file and on the command line, then the one specified on the command line takes precedence.
Restrictions:	None

User Migration Utility Usage Examples

The following sections contain examples of the syntax for some typical uses of this utility.

Migrating Users While Retaining Their Own Schemas

To migrate users while retaining their old database schemas, set the MAPSCHEMA parameter to PRIVATE, which is the default setting. For example, to migrate users scott1, scott2, and all external database users, retaining their old schemas, to the directory at c=Users, c=us with the newly generated database and directory passwords, use the syntax shown in Example A–2.

> **Note:** All external users being migrated are considered non-Kerberos by default. For existing Kerberos users, you can have the utility set their Kerberos principal name attribute in Oracle Internet Directory after migration. To do this, specify the KREALM parameter on the command line by using the Kerberos REALM value. For example, if the Kerberos REALM value is ACME.COM, then you would enter KREALM=ACME.COM. Once you do this, those users with names of the form *user@kerberos_realm* are considered Kerberos users. In Oracle Internet Directory, their Kerberos principal names are set by using their database user names.
>
> See Also: "Keyword: KREALM" on page A-12

Example A–2   Migrating Users with MAPSCHEMA=PRIVATE (Default)

```
umu PHASE=ONE
DBLOCATION=machine1:1521:ora_sid
DBADMIN=system:manager
USERS=ALL_EXTERNAL:LIST
USERSLIST=scott1:scott2
DIRLOCATION=machine2:636
CONTEXT="c=Users,c=us"
ENTADMIN="cn=janeadmin":welcome

umu PHASE=TWO
DBLOCATION=machine1:1521:ora_sid
DBADMIN=system:manager
DIRLOCATION=machine2:636
ENTADMIN="cn=janeadmin":welcome
```

After Phase One is completed successfully, the interface table is populated with the user migration information. Then, the enterprise user administrator can review the table to confirm its contents. Because no value was specified for the MAPSCHEMA parameter, the utility runs Phase One using the default value, PRIVATE, so all users' old database schemas and objects are retained.

Migrating Users and Mapping to a Shared Schema

To migrate users and map them to a new shared schema, dropping their old database schemas, set the MAPSCHEMA parameter to SHARED. The shared schema must already exist, or the enterprise user administrator must create it before running the utility with this parameter setting. In the following example, users scott1, scott2, and all external database users are migrated to the directory at c=Users, c=us with newly

generated database and directory passwords, while mapping all migrated users to a new shared schema in the database.

Use the syntax shown in Example A–3 to run the migration process with MAPSCHEMA set to SHARED.

Example A–3   Migrating Users with MAPSCHEMA=SHARED

```
umu PHASE=ONE
DBLOCATION=machine1:1521:ora_sid
DBADMIN=system:manager
USERS=ALL_EXTERNAL:LIST
USERSLIST=scott1:scott2
MAPSCHEMA=SHARED:schema_32
DIRLOCATION=machine2:636
CONTEXT="c=Users, c=us"
ENTADMIN="cn=janeadmin":welcome

umu PHASE=TWO
DBLOCATION=machine1:1521:ora_sid
DBADMIN=system:manager
DIRLOCATION=machine2:636
ENTADMIN="cn=janeadmin":welcome
```

After Phase One is completed successfully, the interface table is populated with the user migration information. Then, the administrator can review the table to confirm its contents. Users scott1, scott2, and the external users are assigned new randomly generated database and directory passwords. Because no value was specified for the CASCADE parameter, the utility runs Phase One using the default value, NO, which means that migrating users who own database objects in their old database schemas will fail and their schemas will not be automatically dropped. To determine which users have failed, review the log file that is located at $ORACLE_ HOME/network/log/umu.log by default.

Mapping Users to a Shared Schema Using Different CASCADE Options

The CASCADE parameter setting determines whether users' old database schemas are automatically dropped when mapping to a shared schema during migration. CASCADE can be used only when MAPSCHEMA is set to SHARED.

Mapping Users to a Shared Schema with CASCADE=NO

By default, the CASCADE parameter is set to NO. This setting means that when mapping migrating users to a shared schema, users who own database objects in their old schemas are not migrated. For users who do not own database objects, their old database schemas are automatically dropped, and they are mapped to the new shared schema.

> **See Also:**   Example A–3 on page A-16 for a syntax example to map users to a shared schema with CASCADE set to NO. Note that because NO is the default setting for CASCADE, this parameter does not have to be specified in the utility command syntax

Mapping Users to a Shared Schema with CASCADE=YES

If it is known that no migrating users own database objects or want to retain the objects that they own in their old database schemas, then setting the CASCADE parameter to YES automatically drops all users' schemas and schema objects and maps them to the new shared schema. Example A–4 shows the syntax to use when setting

CASCADE to YES. In this example, users scott1, scott2, and all external database users are migrated to the directory at c=Users, c=us, while mapping all migrating users to a new shared schema in the database.

Example A–4   Migrating Users with Shared Schema Mapping and CASCADE=YES

```
umu PHASE=ONE
DBLOCATION=machine1:1521:ora_sid
DBADMIN=system:manager
USERS=ALL_EXTERNAL:LIST
USERSLIST=scott1:scott2
MAPSCHEMA=SHARED:schema_32
CASCADE=YES
DIRLOCATION=machine2:636
CONTEXT="c=Users, c=us"
ENTADMIN="cn=janeadmin":welcome

umu PHASE=TWO
DBLOCATION=machine1:1521:ora_sid
DBADMIN=system:manager
DIRLOCATION=machine2:636
ENTADMIN="cn=janeadmin":welcome
```

After Phase One is completed successfully, the interface table is populated with the user migration information. Then, the administrator can review the table to confirm its contents. Because the CASCADE parameter is set to YES, all migrated users' old database schemas are automatically dropped, including those who own database objects.

> **Caution:**   If you set the CASCADE parameter to YES, then Oracle recommends that enterprise user administrators back up the database or take an export dump of the users being migrated before running this utility. Then, if migrated users want their old database objects, then they can retrieve them from the export dump.

Mapping Users to a Shared Schema Using Different MAPTYPE Options

When MAPSCHEMA is set to SHARED, the mapping type can be set by specifying a value for the MAPTYPE parameter. This parameter takes two values, which are mapping type and mapping level.

Mapping type can be set at DB, for database, or DOMAIN, for enterprise domain. When mapping type DB is specified, the mapping is applied only to the database where the shared schema is stored. When DOMAIN is specified as the mapping type, the mapping is applied to the enterprise domain that contains the database where the shared schema is stored and also applies to all databases in that domain.

Mapping level can be set to ENTRY or SUBTREE. When ENTRY is specified, users are mapped to the shared schema using their full distinguished name (DN). This results in one mapping for each user. When SUBTREE is specified, groups of users who share part of their DNs are mapped together. This results in one mapping for user groups already grouped under some common root in the directory tree. Example A–5 shows the syntax to use when using the MAPTYPE parameter. In this example, users scott1, scott2, and all external database users are migrated to the directory at c=Users, c=us, while mapping all migrated users to a new shared schema in the database. In this example, the mapping will apply to the enterprise domain that contains the

database, and the mapping will be performed at the entry level, resulting in a mapping for each user.

Example A–5   Migrating Users with Shared Schema Mapping Using the MAPTYPE Parameter

```
umu PHASE=ONE
DBLOCATION=machine1:1521:ora_sid
DBADMIN=system:manager
USERS=ALL_EXTERNAL:LIST
USERSLIST=scott1:scott2
MAPSCHEMA=SHARED:schema_32
MAPTYPE=DOMAIN:ENTRY
DIRLOCATION=machine2:636
CONTEXT="c=Users, c=us"
ENTADMIN="cn=janeadmin":welcome

umu PHASE=TWO
DBLOCATION=machine1:1521:ora_sid
DBADMIN=system:manager
DIRLOCATION=machine2:636
ENTADMIN="cn=janeadmin":welcome
```

About Using the SUBTREE Mapping Level Option If a user (`scott`, for example) who is being migrated will have future user entries in a subtree under it, then it makes sense to create a subtree level mapping from this user entry (`cn=scott`) to a schema. However, the database does not interpret the user to be in the subtree so the mapping does not apply to `scott` himself. For example, if you are migrating the user `scott` with the DN `cn=scott,o=acme`, and you choose `SUBTREE` as the mapping level when you run the utility, then a new mapping is created from `cn=scott,o=acme` to the shared schema, but the user `scott` is not mapped to that schema. Only new users who are created under the `scott` directory entry are mapped to the shared schema. Consequently, the `SUBTREE` mapping level should only be specified when user directory entries are placed under other user directory entries, which would be an unusual directory configuration.

If you want an arbitrary subtree user to be mapped to a single shared schema with only one mapping entry, then you must use Enterprise Security Manager to create that mapping.

> **See Also:**   "Managing Enterprise Domain Database Schema Mappings" on page 4-13 for information about using Enterprise Security Manager

Migrating Users Using the PARFILE, USERSFILE, and LOGFILE Parameters

It is possible to enter user information and User Migration Utility parameters into a text file and pass the information and parameters to the utility using the `PARFILE` and `USERSFILE` parameters. The `LOGFILE` parameter sets the directory path for the log file where details about the migration for each user are written.

The `PARFILE` parameter tells the utility where a text file is located that contains the parameters for a bulk user migration. The `USERSFILE` parameter works like the `PARFILE` parameter, except that it contains database users instead of parameters. The parameters and users lists contain one parameter or user for each line. The `LOGFILE` parameter tells the utility where to write the system events that occur during a user migration, such as errors. Use the `USERSFILE` parameter during Phase One of the

migration process. The `PARFILE` and `LOGFILE` parameters can be used in both phases.

Example A–6 shows the syntax for a typical parameter text file to migrate users `scott1`, `scott2`, and all external database users, while retaining their old schemas, to the directory at `c=Users,  c=us`. In this example, a log of migration events is written to the file `errorfile1` in the directory where the utility is run. If another location is desired, then include the path with the file name.

Example A–6   Parameter Text File (par.txt) to Use with the PARFILE Parameter

```
DBLOCATION=machine1:1521:ora_sid
DBADMIN=system:manager
USERS=ALL_EXTERNAL:LIST:FILE
USERSLIST=scott1:scott2
USERSFILE=usrs.txt
DIRLOCATION=machine2:636
CONTEXT="c=Users, c=us"
ENTADMIN="cn=janeadmin":welcome
LOGFILE=errorfile1
```

Example A–7 shows the syntax for a typical users list text file.

Example A–7   Users List Text File (usrs.txt) to Use with the USERSFILE Parameter

```
user1
user2
user3
```

To run Phase One of the migration process with these parameters and users list text files, use the syntax shown in Example A–8.

Example A–8   Migrating Users Using the PARFILE, USERSFILE, and LOGFILE Parameters

```
umu PHASE=ONE
DBADMIN=system:manager
PARFILE=par.txt
LOGFILE=errorfile2
```

> **Note:**   Although the `LOGFILE` parameter is specified twice, once in the parameter text file as `errorfile1` (shown in Example A–6) and once on the command line as `errorfile2` (shown in Example A–8), command-line parameters take precedence over those specified inside the parameter file. Consequently, in Example A–8, the log file will be written to `errorfile2` because that value is specified on the command line.

Troubleshooting Using the User Migration Utility

Migration failures are reported to the enterprise user administrator with error messages and log messages. The following sections describe common error and log messages and what administrators can do to resolve them.

> **See Also:**   "Summary of User Migration Utility Error and Log Messages" on page A-26 for an alphabetical listing of error and log messages and links to where they are described in this section

Common User Migration Utility Error Messages

When the utility encounters any error while running, it displays an error message and stops running. The following sections describe these messages and explain how to resolve the errors:

- Resolving Error Messages Displayed for Both Phases
- Resolving Error Messages Displayed for Phase One

Resolving Error Messages Displayed for Both Phases

The following error messages may be displayed while the utility is running either Phase One or Phase Two of the migration:

- Attribute value missing : : orclCommonNicknameAttribute
- Database connection failure
- Database error: < database_error_message >
- Database not in any domain : : DB-NAME = < database_name >
- Database not registered with the directory : : DB-NAME = < dbName >
- Directory connection failure
- Directory error : : < directory_error_message >
- Multiple entries found : : uniqueMember = < database_DN >

Attribute value missing : : orclCommonNicknameAttribute

Cause: The nickname attribute is not set in the directory in the root identity management realm.

Action: Use Enterprise Security Manager Console to set the nickname attribute for the identity management realm.

Database connection failure

Cause: The utility was unable to connect to the database.

Action: Perform these steps:

1. Check the database status to determine whether it is configured for encryption and integrity.

2. Check the privileges and credentials of the enterprise user administrator who is running the utility.

Database error: < database_error_message >

Cause: The utility encountered a database error.

Action: Check the database error message details for the database.

> **See Also:** *Oracle Database Error Messages* for information about resolving database error messages

Database not in any domain : : DB-NAME = < database_name >

Cause: The database is not a member of any enterprise domain.

Action: Use Enterprise Security Manager to add the database to an enterprise domain in the directory.

Database not registered with the directory : : DB-NAME = < dbName >

Cause: There is no entry for the database in the Oracle Context that the ldap.ora file points to.

Action: Use Database Configuration Assistant or Enterprise Security Manager to register the database in the directory.

Directory connection failure

Cause: The utility was unable to connect to the directory.

Action: Perform these steps:

1. Check the directory server status to determine whether the directory server port is configured for SSL with no authentication.

2. Check the privileges and credentials of the enterprise user administrator who is running the utility.

Directory error : : < directory_error_message >

Cause: The utility encountered a directory error.

Action: Check the directory error message details for the directory.

> **See Also:** *Oracle Internet Directory Administrator's Guide* information about resolving error messages for Oracle Internet Directory

Multiple entries found : : uniqueMember = < database_DN >

Cause: The database belongs to more than one enterprise domain in the directory.

Action: Use Enterprise Security Manager or Oracle Directory Manager to ensure that the database belongs to only one enterprise domain.

Resolving Error Messages Displayed for Phase One

While the utility is running Phase One of the migration, syntax or other types of errors may occur. The following error messages may be displayed while the utility is running Phase One of the migration:

- Argument missing or duplicated : :
- Database object missing : : SHARED-SCHEMA = <shared_schema_name >
- Error reading file : : < file_name > : : < io_error_message >
- Error reading file : : PARFILE = < file_name > : : < io_error_message>
- Getting local host name failed
- Interface table creation in SYS schema not allowed
- Invalid argument or value : : < argument >
- Invalid arguments for the phase
- Invalid value : : < user > [USERSFILE]
- Invalid value : : < user > [USERSFILE] { = = DBADMIN }
- Invalid value : : < user > [USERSLIST]
- Invalid value : : < user > [USERSLIST] { = = DBADMIN }
- Logging failure : : < io_error_message >
- No entry found : : CONTEXT = < context >

Argument missing or duplicated : : < parameter >

Cause: Syntax error. A parameter is missing or has been entered multiple times.

Action: Check the usage syntax.

Database object missing : : SHARED-SCHEMA = <shared_schema_name >

Cause: The shared schema is not present in the database.

Action: Create the shared schema.

Error reading file : : < file_name > : : < io_error_message >

Cause: Syntax error. The utility cannot read the file that contains the users list that is specified in the USERSFILE parameter.

Action: Perform these steps:

1. Check to ensure that the file exists.

2. Check to ensure that the file has the correct permissions so the utility can read it.

Error reading file : : PARFILE = < file_name > : : < io_error_message>

Cause: Syntax error. The utility cannot read the file that contains the list of parameters that is specified in the PARFILE parameter.

Action: Perform these steps:

1. Check to ensure that the file exists.

2. Check to ensure that the file has the correct permissions so the utility can read it.

Getting local host name failed

Cause: Syntax error. The utility is unable to read the local host name for the database location or the directory location.

Action: Explicitly enter the host name information with the DBLOCATION and DIRLOCATION parameters.

> **See Also:**
>
> - "Keyword: DBLOCATION" on page A-9
> - "Keyword: DIRLOCATION" on page A-9
>
> for information about how to use these parameters

Interface table creation in SYS schema not allowed

Cause: The interface table cannot be created in the SYS schema.

Action: Specify another user in the DBADMIN parameter.

> **See Also:** "Keyword: DBADMIN" on page A-10 for information about setting the DBADMIN parameter

Invalid argument or value : : < argument >

Cause: Syntax error. The argument name or value has been entered incorrectly.

Action: Check the usage syntax.

See Also:

- "User Migration Utility Command-Line Syntax" on page A-7
- "Accessing Help for the User Migration Utility" on page A-8
- "User Migration Utility Parameters" on page A-8

for information about using the command-line syntax for this utility

Invalid arguments for the phase

Cause: Syntax error. This occurs when you have used a command-line argument that is only intended for Phase One, but you are running Phase Two.

Action: Check the usage syntax.

Invalid value : : < user > [USERSFILE]

Cause: Syntax error. The user that is specified in this error message is invalid because he is not a user in the database that is specified in the DBLOCATION parameter.

Action: Remove the invalid user from the file that is specified with the USERSFILE parameter.

Invalid value : : < user > [USERSFILE] { = = DBADMIN }

Cause: Syntax error. The file that is specified in the USERSFILE parameter contains the user who is running the migration utility.

Action: Remove that user from the file.

Invalid value : : < user > [USERSLIST]

Cause: Syntax error. The user that is specified in this error message is invalid because they are not a user in the database that is specified in the DBLOCATION parameter.

Action: Remove the invalid user from the USERSLIST parameter.

Invalid value : : < user > [USERSLIST] { = = DBADMIN }

Cause: Syntax error. The USERSLIST parameter contains the user who is running the migration utility.

Action: Remove that user from the USERSLIST parameter.

Logging failure : : < io_error_message >

Cause: Syntax error. The utility cannot find the log file or it cannot open the file to write to it.

Action: Perform these steps:

1. Check to ensure that the log file exists.

2. Check to ensure that the log file has the correct permissions so the utility can write information to it.

No entry found : : CONTEXT = < context >

Cause: The CONTEXT entry is not present in the directory.

Action: Perform one of the following steps:

- Use the directory management tool or the LDAP command-line utility to create an entry in the directory for the context value.

■ Specify another valid context value.

Resolving Error Messages Displayed for Phase Two

Most of the error messages that you encounter while running this utility occur in Phase One. After Phase One has completed successfully, and while Phase Two is running, the following error may occur:

Database object missing : : TABLE = ORCL_GLOBAL_USR_MIGRATION_DATA

Cause: The utility cannot find the interface table.

Action: Perform one of the following steps:

■ Run Phase One of the utility to create the interface table.

■ Check to ensure that the user who is specified in the DBADMIN parameter is the same user who was specified for that parameter for Phase One.

Common User Migration Utility Log Messages

Typically, log messages are written to the log file for each user who is migrated, whether the user was migrated successfully or not. The following sections describe these messages and explain how to resolve the errors:

Common Log Messages for Phase One

While the utility is running Phase One of the migration, messages that indicate that a user's information has not been successfully populated in the interface table may be written to the log file. After the utility completes Phase One, review the log file to check for the following messages:

■ Multiple entries found : : < nickname_attribute > = < username >

■ No entry found : : < nickname_attribute > = < username > : : Entry found : DN = < dn >

Multiple entries found : : < nickname_attribute > = < username >

Cause: The nickname attribute matches multiple users or the user matches with multiple nickname attributes.

Action: Resolve the multiple matches and run the utility again for the users whose log file entry displayed this message.

No entry found : : < nickname_attribute > = < username > : : Entry found : DN = < dn >

Cause: No entry was found for the nickname matching, but an entry already exists for the DN in the directory.

Action: Specify a different DN for the user.

Common Log Messages for Phase Two

While the utility is running Phase Two of the migration, messages that indicate that a user has not successfully migrated may be written to the log file. After the utility completes Phase Two, review the log file to check for the following messages:

■ Attribute exists : : orclPassword

■ Attribute value missing : : orclPassword

■ Database object missing : : SHARED-SCHEMA = < shared_schema >

■ Entry found : : DN = < user_DN >

- Invalid value : : <interface_table_column_name> = < interface_table_column_ value >

- No entry found : : DN = < user_DN >

Attribute exists : : orclPassword

This message typically occurs with the message `Invalid value::<column_ name>=<column_value>`.

Cause: The entry already contains a value for the `orclPassword` attribute.

Action: Check the `DBPASSWORD_EXIST_FLAG` column in the interface table for a `T/F` value that correctly reflects whether a database password exists for this user.

Attribute value missing : : orclPassword

This message typically occurs with the message `Invalid value::<column_ name>=<column_value>`.

Cause: The `orclPassword` attribute of this user's entry has a null value.

Action: Check the `DBPASSWORD_EXIST_FLAG` column in the interface table for a `T/F` value that correctly reflects whether a database password exists for this user.

Database object missing : : SHARED-SCHEMA = < shared_schema >

Cause: The shared schema that was specified for this user does not exist in the database.

Action: Perform one of the following steps:

- Check to ensure that the correct shared schema was specified for this user. If the shared schema name was incorrectly specified, then edit the `SHARED_ SCHEMA` column of the interface table and run Phase Two of the utility for this user again.

- Create the shared schema in the database and run Phase Two of the utility for this user again.

Entry found : : DN = < user_DN >

This message typically occurs with the message `Invalid value::<column_ name>=<column_value>`.

Cause: An entry already exists for the specified user DN.

Action: Check the `USERDN_EXIST_FLAG` column in the interface table for a `T/F` value that correctly reflects whether a user entry already exists in the directory for this DN.

Invalid value : : <interface_table_column_name> = < interface_table_column_value >

Cause: The value in the interface table column for this user is invalid. Typically, this message is accompanied by additional log messages for this user.

Action: Check to ensure that the correct value has been entered for this user.

No entry found : : DN = < user_DN >

This message typically occurs with the message `Invalid value::<column_ name>=<column_value>`.

Cause: The entry for the DN is missing in the directory.

Action: Check the `USERDN_EXIST_FLAG` column in the interface table for a `T/F` value that correctly reflects whether a user entry already exists in the directory for this DN.

Summary of User Migration Utility Error and Log Messages

Table A–4 and Table A–5 list all of the error and log messages in alphabetical order and provides links to the section in this chapter that describes the message and how to resolve it.

Table A–4   Alphabetical Listing of User Migration Utility Error Messages

User Migration Utility Error Message	Phase
Argument missing or duplicated : : < parameter > on page A-22	1
Attribute value missing : : orclCommonNicknameAttribute on page A-20	Both
Database connection failure on page A-20	Both
Database error: < database_error_message > on page A-20	Both
Database not in any domain : : DB-NAME = < database_name > on page A-20	Both
Database not registered with the directory : : DB-NAME = < dbName > on page A-20	Both
Database object missing : : SHARED-SCHEMA = <shared_schema_name > on page A-22	1
Database object missing : : TABLE = ORCL_GLOBAL_USR_MIGRATION_DATA on page A-24	2
Directory connection failure on page A-21	Both
Directory error : : < directory_error_message > on page A-21	Both
Error reading file : : < file_name > : : < io_error_message > on page A-22	1
Error reading file : : PARFILE = < file_name > : : < io_error_message> on page A-22	1
Getting local host name failed on page A-22	1
Interface table creation in SYS schema not allowed on page A-22	1
Invalid argument or value : : < argument > on page A-22	1
Invalid arguments for the phase on page A-23	1
Invalid value : : < user > [USERSFILE] on page A-23	1
Invalid value : : < user > [USERSFILE] { = = DBADMIN } on page A-23	1
Invalid value : : < user > [USERSLIST] on page A-23	1
Invalid value : : < user > [USERSLIST] { = = DBADMIN } on page A-23	1
Logging failure : : < io_error_message > on page A-23	1
Multiple entries found : : uniqueMember = < database_DN > on page A-21	Both
No entry found : : CONTEXT = < context > on page A-23	1

Table A–5   Alphabetical Listing of User Migration Utility Log Messages

User Migration Utility Log Message	Phase
Attribute exists : : orclPassword on page A-25	2
Attribute value missing : : orclPassword on page A-25	2
Database object missing : : SHARED-SCHEMA = < shared_schema > on page A-25	2
Entry found : : DN = < user_DN > on page A-25	2
Invalid value : : <interface_table_column_name> = < interface_table_column_value > on page A-25	2

Table A–5   (Cont.) Alphabetical Listing of User Migration Utility Log Messages

User Migration Utility Log Message	Phase
Multiple entries found : : < nickname_attribute > = < username > on page A-24	1
No entry found : : DN = < user_DN > on page A-25	2
No entry found : : < nickname_attribute > = < username > : : Entry found : DN = < dn > on page A-24	1

B

SSL External Users Conversion Script

You should run the SSL external users conversion script after upgrading to Oracle Database 10*g* Release 2 (10.2), in case you were using SSL-authenticated external users in the earlier release. The script converts SSL-authenticated external users in previous releases into SSL-authenticated external users in Oracle Database 10*g* Release 2 (10.2).

This chapter covers the following topics:

- Using the SSL External Users Conversion Script
- Converting Global Users into External Users

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
--dbuserpassword database user password
[-a]
[-l username1,username2,...]
[-f filename]
[-o]
[-h]
```

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.

Use the `-a` option to convert all SSL-authenticated external users. Here is an example:

```
extusrupgrade --dbconnectstring dlsun88:1521:10gR2 --dbuser system
--dbuserpassword manager -a
```

Use the `-l` option to specify a comma-separated list of users to be converted. For example:

```
extusrupgrade --dbconnectstring dlsun88:1521:10gR2 --dbuser system
--dbuserpassword manager -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 dlsun88:1521:10gR2 --dbuser system
--dbuserpassword manager -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 dlsun88:1521:10gR2 --dbuser system
--dbuserpassword manager -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.

Converting Global Users into External Users

Oracle Database 10*g* Release 2 (10.2) 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.

For example:

```
extusrupgrade --dbconnectstring dlsun88:1521:10gR2 --dbuser system
--dbuserpassword manager -l user1,user2
```

The preceding example converts two global users into external users.

C

Integrating Enterprise User Security with Microsoft Active Directory

Enterprise users 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 Active Directory to store and manage user entries, then you can integrate it with Oracle Internet Directory to manage Enterprise User Security.

Kerberos authentication for enterprise users can make use of tickets issued by a kerberos Key Distribution Center (KDC) running on a Microsoft Windows domain controller.

This appendix lists the steps involved in integrating Enterprise User Security with Microsoft Active Directory using kerberos for authentication. It includes the following sections:

- Set Up Synchronization Between Active Directory and Oracle Internet Directory
- Set Up a Windows 2000 Domain Controller to Interoperate with Oracle Client
- Set Up Oracle Database to Interoperate with a Windows 2000 Domain Controller
- Set Up Oracle Database Client to Interoperate with a Windows 2000 KDC
- Obtain an Initial Ticket for the Client
- Configure Enterprise User Security for Kerberos Authentication

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 Active Directory for centralized administration. You should set up synchronization between Oracle Internet Directory and Active Directory before you configure Enterprise User Security to work with 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*

Set Up a Windows 2000 Domain Controller to Interoperate with Oracle Client

The following tasks must be performed on the Windows 2000 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 `ktpass` command-line utility to create a kerberos `keytab` file

   The `ktpass` utility is a part of the Windows 2000 Support Tools. The `keytab` file is required to use a Windows 2000 domain controller as a KDC.

3. Copy the `keytab` file created in the previous step to the computer on which the database server is installed

   > **See Also:** *Oracle Database Advanced Security Administrator's Guide* for a detailed listing of the preceding steps.

Set Up Oracle Database to Interoperate with a Windows 2000 Domain Controller

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 Advanced Security Administrator's Guide* for a detailed description of the preceding step.

Set Up Oracle Database Client to Interoperate with a Windows 2000 KDC

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 Windows 2000 domain controller 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 Advanced Security Administrator's Guide* for a detailed listing of the preceding steps.

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 Advanced Security Administrator's Guide* for more details on requesting an initial ticket with `okinit`.

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 the Enterprise Security 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" on page 3-16.

D

Upgrading from Oracle9i to Oracle Database 10*g* Release 2 (10.2)

This appendix discusses upgrading Oracle9*i* Database to Oracle Database 10*g* Release 2 (10.2) with respect to Enterprise User Security. It includes the following sections:

- Upgrading Oracle Internet Directory from Release 9.2 to Release 9.0.4
- Upgrading Oracle Database from Release 9.2 to Release 10.2

Upgrading Oracle Internet Directory from Release 9.2 to Release 9.0.4

Oracle9*i* Database Release 2 can work with Oracle Internet Directory Release 9.2 or Release 9.0.4. Oracle Database 10*g* Release 2 (10.2) requires Oracle Internet Directory 9.0.4 or later. In case you are using Oracle Internet Directory Release 9.2, you need to upgrade it to Release 9.0.4.

The following list discusses upgrading Oracle Internet Directory Release 9.2 to Oracle Internet Directory Release 9.0.4:

1. Use Oracle Internet Directory Configuration Assistant to upgrade Oracle Internet Directory. This is required if you want to register Oracle Database 10*g* Release 2 (10.2) instances in the directory.

2. Upgrade Oracle Contexts used for Enterprise User Security to Identity Management Realms, if they are not root contexts. Use the Oracle Internet Directory Configuration Assistant command-line utility as follows:

   ```
   oidca mode=CTXTOIMR
   ```

   This step is required if you want to register an Oracle Database 10*g* Release 2 (10.2) instance in a realm.

   You cannot use the root Oracle Context for Oracle Database 10*g* Release 2 (10.2) databases because it is not an Identity Management Realm.

3. Use Oracle Internet Directory tools, such as `ldapmodify` and `bulkmodify`, to add the `orcluserV2 objectclass` to existing user entries. This `objectclass` is required for users to change their database passwords, and for kerberos authentication to the database.

4. In a realm that contains both Oracle9*i* Database (Release 9.1 or Release 9.2) and Oracle Database 10*g* Release 2 (10.2), use a DAS-based tool in Oracle Internet Directory Release 9.0.4 to create and manage users. You can use either Oracle Internet Directory Self-Service Console or Enterprise Security Manager Console. Do not use Enterprise Security Manager or Enterprise Login Assistant from Oracle9*i* installations.

Upgrading Oracle Database from Release 9.2 to Release 10.2

For each Oracle9*i* Database instance that you upgrade to Oracle Database 10*g* Release 2 (10.2), perform the following steps:

1. Use Oracle Wallet Manager to disable automatic login for the database wallet.

2. Copy the database distinguished name (DN) from the initialization parameter `rdbms_server_dn` to a file in a secure location.

3. Upgrade the database to Oracle Database 10*g* Release 2 (10.2).

4. Depending on where your database `admin` directory is stored, move the database wallet either to `$ORACLE_BASE/admin/`*olddbuniquename*`/wallet` or `$ORACLE_HOME/admin/`*olddbuniquename*`/wallet`. Note that `$ORACLE_HOME` is for the new Oracle Database 10*g* Release 2 (10.2). You may have to create the `wallet` directory.

5. Copy the old `$ORACLE_HOME/network/admin/ldap.ora` file to the new `$ORACLE_HOME/ldap/admin/ldap.ora` file. Alternatively, you can use Oracle Net Configuration Assistant to create a new `ldap.ora` file.

6. Use the command-line utility, `mkstore`, to put the database DN (from the file in the previously created secure directory location) into the wallet by using the following syntax:

   ```
   mkstore -wrl database_wallet_location -createEntry
   ORACLE.SECURITY.DN database_DN
   ```

   You will be prompted for the wallet password.

   If you make a mistake in the `mkstore` command, then you can use the `-modifyEntry` option to correct it.

7. Use Database Configuration Assistant to generate the database-to-directory password in the database wallet. Choose the Modify Database option.

8. Use Oracle Wallet Manager to re-enable automatic login for the database wallet.

9. Use Oracle Net Manager to set the new wallet location in the `sqlnet.ora` file to the directory specified in step 4.

The default for the nickname attribute, such as CN, remains unchanged. The upgrade process does not change the default nickname attribute setting. After upgrading from Oracle Internet Directory Release 9.2 to Release 9.0.4, if you are unable to log into Oracle Database 10*g* Release 2 (10.2), then you must use the DAS-based Oracle Internet Directory Self-Service Console to reset your password.

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/password@connect_identifier
```

connect string

Information the user passes to a **service** to connect, such as **username**, password and **net service name**. For example:

```
CONNECT username/password@net_service_name
```

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.

Data Encryption Standard (DES)

The U.S. data encryption standard.

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).

DES

See **Data Encryption Standard (DES)**

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.

Diffie-Hellman key negotiation algorithm

This is a method that lets two parties communicating over an insecure channel to agree upon a random number known only to them. Though the parties exchange information over the insecure channel during execution of the Diffie-Hellman key negotiation algorithm, it is computationally infeasible for an attacker to deduce the random number they agree upon by analyzing their network communications. Oracle Advanced Security uses the Diffie-Hellman key negotiation algorithm to generate session keys.

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 networkservices; 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 role**s. 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 user**s. A set of Oracle role-based **authorization**s across one or more databases in an **enterprise domain**. Enterprise roles are stored in the directory and contain one or more **global role**s.

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* 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 machine 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**.

MD5

An algorithm that assures data integrity by generating a 128-bit cryptographic message digest value from given data. If as little as a single bit value in the data is modified, the MD5 checksum for the data changes. Forgery of data in a way that will cause MD5 to generate the same result as that for the original data is considered computationally infeasible.

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 machine, 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 domain**s 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, **attribute**s, **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 is slightly slower than MD5, but the 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 machine 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