

Oracle9i

Directory Service Integration and Deployment Guide

Release 2 (9.2)

March 2002

Part No. A96579-01

ORACLE®

Part No. A96579-01

Copyright © 2002 Oracle Corporation. All rights reserved.

Primary Author: Henry Abrecht

Graphic Artist: Valarie Moore

Contributors: Deanna Bradshaw, Torrance Brooksfuller, Kristy Browder, Montgomery Close, Michael Cowan, Cheng Han, Marilyn Hollinger, Cynthia Kibbe, Ashish Kolli, Nina Lewis, Michael Mesaros, Janaki Narasinghanallur, David Saslav, Daniele Schechter, Richard Smith, Uppili Srinivasan, Deborah Steiner, Rama Vissapragada, Wei Wang, Rodney Ward, and Daniel Wong

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle Names, Oracle Press, Oracle Store, Oracle8i, Oracle9i, PL/SQL, and SQL*Plus are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	xiii
Preface.....	xv
Audience	xvi
Organization.....	xvi
Related Documentation	xvii
Conventions.....	xviii
Documentation Accessibility	xxiv
1 The Oracle Directory Environment	
The Importance of LDAP to Oracle Products.....	1-2
What Directories Do Oracle Products Work With?	1-2
LDAP-Enabled Oracle Products.....	1-2
Oracle Internet Directory	1-4
Integrating Oracle Products with Third-Party Directories.....	1-5
2 Directory Server Concepts	
What Is a Directory?	2-2
What Is an Online Directory?.....	2-2
Directories and Relational Databases Compared.....	2-2
Read-to-Write Ratio	2-3
Data Units.....	2-3

Distribution	2-3
Entries.....	2-4
Typical Directory Applications	2-4
The Benefits of Standards-Based Online Directories	2-4
What Is LDAP?	2-5
The Benefits of LDAP	2-5
LDAP Version 3	2-6
C LDAP API.....	2-7
LDIF	2-7
Directory Information Flow: An Example.....	2-8
Directory Components and Features.....	2-9
Entries	2-9
Attributes	2-11
Types of Attributes	2-11
Attribute Syntaxes and Matching Rules.....	2-12
Foreign-Language Options for Attributes	2-13
Object Classes.....	2-13
Types of Object Classes.....	2-14
Structural Object Classes	2-14
Auxiliary Object Classes.....	2-14
Abstract Object Classes.....	2-15
Creating New Object Classes and Redefining Old Ones.....	2-15
Naming Contexts	2-15
Schema.....	2-16
Security.....	2-17
Authentication	2-17
Access Control Lists	2-17
The Oracle Context	2-19

3 Planning and Deployment Guidelines

What Do You Put in a Directory?	3-2
Designing an Effective Directory Tree and Choosing Entry Names.....	3-2
Physically Distributing the Directory: Partitions and Replicas	3-3
Reasons to Replicate	3-3
Reasons to Partition.....	3-4

Designing for High Availability and Failover	3-4
Capacity Planning, Sizing, and Tuning	3-5
Capacity Planning	3-5
Sizing	3-5
Tuning	3-6
Designing Directory Security	3-7

4 Deploying Oracle Products with Oracle Internet Directory

Oracle Net Services	4-2
How Oracle Net Services Uses Oracle Internet Directory	4-3
Oracle Net Services Entries Under the Oracle Context	4-5
Security Measures for Oracle Net Services Entries	4-9
Directory Deployment Factors for Oracle Net Services	4-9
Oracle Advanced Security	4-11
How Oracle Advanced Security Uses Oracle Internet Directory	4-11
Central Management of User Authentication Credentials	4-11
Central Management of User Authorizations	4-11
Mappings to Shared Schemas	4-12
Single Password Authentication	4-12
Single Sign-On	4-12
Central Storage of PKI Credentials	4-12
Oracle Advanced Security Entries Under the Oracle Context	4-12
Security Measures for Oracle Advanced Security Entries	4-14
Directory Deployment Factors for Oracle Advanced Security	4-15
Application Context	4-16
How Application Context Uses Oracle Internet Directory	4-16
Application Context Entries Under the Oracle Context	4-17
Security Measures for Application Context Entries	4-19
Oracle Advanced Queuing	4-20
How Oracle Advanced Queuing Uses Oracle Internet Directory	4-20
Oracle Advanced Queuing Entries Under the Oracle Context	4-21
Security Measures for Oracle Advanced Queuing Entries	4-22
Directory Deployment Factors for Oracle Advanced Queuing	4-22
Oracle Dynamic Services	4-23
How Oracle Dynamic Services Uses Oracle Internet Directory	4-25

Oracle Dynamic Services Entries Under the Oracle Context	4-28
Security Measures for Oracle Dynamic Services Entries	4-30
Directory Deployment Factors for Oracle Dynamic Services	4-30

5 Completing Directory Usage Configuration

Prerequisites for Directory Use	5-2
Options for Directory Usage Configuration	5-3
Configuring Directory Usage After Installing the Database	5-3
Using Oracle Net Configuration Assistant to Configure Directory Usage	5-3
Using Database Configuration Assistant to Register a Database	5-7
Configuring Directory Usage During a Custom Database Installation	5-8
Administrative Groups	5-9
Configuring Directory Usage During a Client Installation	5-9
Product-Specific Configuration Tasks	5-10

A Oracle-Specific LDAP Schema Extensions

Oracle Net Services	A-2
Structural Object Classes	A-2
Attributes	A-2
Oracle Advanced Security	A-4
Structural Object Classes	A-4
Attributes	A-4
Application Context	A-5
Oracle Advanced Queuing	A-6
Structural Object Classes	A-6
Attributes	A-6
Oracle Dynamic Services	A-8
Structural Object Classes	A-8
Attributes	A-9

B LDAP Command-Line Tools

LDAP Command-Line Tools..... B-2
Optional Arguments for Command-Line Tools..... B-8

Index

List of Figures

2-1	Information Flow for Oracle Internet Directory	2-8
2-2	Directory Information Tree with a Distinguished Name Highlighted.....	2-10
2-3	Naming Contexts and Non-Naming Contexts.....	2-16
2-4	Starter Oracle Context with Added Entries for a Database and a Database Connect Descriptor.....	2-20
4-1	Client Using a Directory Server to Resolve a Connect Identifier	4-4
4-2	Networking Entries.....	4-6
4-3	Example of Networking Entries.....	4-6
4-4	Directory Structure with Two Oracle Contexts	4-7
4-5	Directory Entries Relevant to Oracle Advanced Security	4-13
4-6	Directory Information Tree for Application Context, Showing Attributes for the Context Value	4-18
4-7	Directory Information Tree for Oracle Advanced Queuing	4-21
4-8	LDAP Server Within Oracle Dynamic Services Framework Architecture	4-25
4-9	YahooQuote Service Registration	4-27
4-10	Registry Synchronization Process for a New Dynamic Services Engine Instance....	4-27
4-11	Directory Information Tree for Oracle Dynamic Services, Showing Attribute Types for One Service, Currency	4-29
5-1	Oracle Net Configuration Assistant: Directory Usage Configuration Page	5-4

List of Tables

2-1	Directories Versus Relational Databases.....	2-3
2-2	A Typical Directory Entry with Attributes Shown	2-11
2-3	Attribute Syntaxes and Associated Matching Rules	2-12
2-4	Containers Under the Oracle Context	2-21
4-1	Oracle Net Services LDAP Main Object Classes.....	4-8
4-2	Oracle Net LDAP Derived Object Classes	4-8
4-3	Administrative Groups for Oracle Advanced Security	4-14
4-4	Containers for Global Topics Entries.....	4-21
5-1	Links to Product-Specific Configuration Information	5-10
B-1	Commonly Used Command-Line Options	B-8

Send Us Your Comments

Oracle9i Directory Service Integration and Deployment Guide, Release 2 (9.2)

Part No. A96579-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: infodev_us@oracle.com
- FAX: (650) 506-7227 Attn: Server Technologies Documentation Manager

- Postal service:

Oracle Corporation
Server Technologies Documentation
500 Oracle Parkway, Mailstop 4op11
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Directory Service Integration and Deployment Guide is a starting point for those who want to learn how Oracle products use an LDAP-compliant directory, specifically Oracle Internet Directory. The book is also a starting point for learning how to configure Oracle Internet Directory to use Oracle products.

This preface contains these topics:

- [Audience](#)
- [Organization](#)
- [Related Documentation](#)
- [Conventions](#)
- [Documentation Accessibility](#)

Audience

Directory Service Integration and Deployment Guide is intended for the following readers:

- Directory administrators who want to perform the minimal tasks needed to configure a directory for Oracle products
- Directory administrators and others who want to know how Oracle products use a directory
- Directory administrators and others who want to review or become acquainted with directory concepts

The book assumes no prior knowledge of LDAP, although a basic understanding of the protocol and its purpose are helpful.

Organization

This document contains:

Chapter 1, "The Oracle Directory Environment"

This chapter introduces Oracle directory-enabled products and Oracle Internet Directory. It also examines strategies to integrate Oracle products with third-party directories.

Chapter 2, "Directory Server Concepts"

This chapter describes the function of directories, defines the LDAP protocol, and identifies the components of an online directory.

Chapter 3, "Planning and Deployment Guidelines"

This chapter provides an overview of issues to consider before deploying a directory.

Chapter 4, "Deploying Oracle Products with Oracle Internet Directory"

This chapter describes how specific Oracle products use Oracle Internet Directory. It describes where each product stores its entries and how it protects these entries from unauthorized access. The chapter also discusses deployment factors particular to each product.

Chapter 5, "Completing Directory Usage Configuration"

This chapter describes how to configure access to a directory that is already installed. It also provides links to documents that describe directory configuration tasks particular to each of the Oracle products examined in Chapter 4.

Appendix A, "Oracle-Specific LDAP Schema Extensions"

This appendix lists and describes the object classes and attributes that LDAP-enabled Oracle products use to define entries in Oracle Internet Directory.

Appendix B, "LDAP Command-Line Tools"

This appendix lists and describes six popular command-line tools available through the LDAP C-API.

Related Documentation

For more information, see these Oracle resources:

- *Oracle Advanced Security Administrator's Guide*
- *Oracle Dynamic Services User's and Administrator's Guide*
- *Oracle9i Net Services Administrator's Guide*
- *Oracle8i Networking 101* by Marlene L. Theriault. Oracle Press, 2000
- *Oracle9i Application Developer's Guide - Advanced Queuing*
- *Oracle9i Application Developer's Guide - Fundamentals*

In North America, printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

<http://www.oraclebookshop.com/>

Other customers can contact their Oracle representative to purchase printed documentation.

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://otn.oracle.com/admin/account/membership.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://otn.oracle.com/docs/index.htm>

To access the database documentation search engine directly, please visit

<http://tahiti.oracle.com>

For additional information, see:

- *Understanding and Deploying LDAP Directory Services* by Timothy A. Howes, Mark C. Smith, and Gordon S. Good. Macmillan Technical Publishing, 1999
- <http://www.ietf.org/> for information about the LDAP protocol

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 .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle9i Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> 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.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as 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 open 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.
<i>lowercase italic monospace (fixed-width) font</i>	Lowercase italic monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>Uold_release</i> .SQL where <i>old_release</i> 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
[]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE DISABLE}
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> That we have omitted parts of the code that are not directly related to the example That you can repeat a portion of the code 	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
. . . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fs1/dbs/tbs_01.dbf /fs1/dbs/tbs_02.dbf
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;

Convention	Meaning	Example
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>
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. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;

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 >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - <i>HOME_NAME</i> > 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 file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	c:\winnt\"system32 is the same as C:\WINNT\SYSTEM32

Convention	Meaning	Example
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 <i>command prompt</i> 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.	<pre>C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\" C:\>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)</pre>
<i>HOME_NAME</i>	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 <i>HOME_NAME</i> TNSListener

Convention	Meaning	Example
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default used one of the following names:</p> <ul style="list-style-type: none"> ■ C:\orant for Windows NT ■ C:\orawin98 for Windows 98 <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle. 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\orann, where <i>nn</i> is the latest release number. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> <p>Refer to <i>Oracle9i Database Getting Starting for Windows</i> for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.</p>	Go to the <i>ORACLE_BASE\ORACLE_HOME\rdms\admin</i> directory.

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. Standards will continue to evolve over time, and Oracle Corporation 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 additional information, visit the Oracle Accessibility Program Web site at

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

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, 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, JAWS 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 Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

The Oracle Directory Environment

This chapter introduces Oracle directory-enabled products and Oracle Internet Directory. In addition, it takes a brief look at strategies to integrate the Oracle technology stack with third-party directories.

The chapter covers the following topics:

- [The Importance of LDAP to Oracle Products](#)
- [What Directories Do Oracle Products Work With?](#)
- [LDAP-Enabled Oracle Products](#)
- [Oracle Internet Directory](#)
- [Integrating Oracle Products with Third-Party Directories](#)

The Importance of LDAP to Oracle Products

Oracle and other enterprises increasingly use directories compliant with Lightweight Directory Access Protocol (LDAP) to centralize information storage. This information might consist of user names, passwords, e-mail addresses, and network devices such as printers, or it might determine which users are allowed database access. Centralizing this information reduces the need to manage it on multiple databases.

What Directories Do Oracle Products Work With?

Many Oracle products are currently certified to work with Oracle Internet Directory. In addition, work is underway on strategies to use Oracle Internet Directory to provide interoperability between the entire Oracle technology stack and selected third-party directories. By addressing the entire technology stack, instead of individual components, interoperability and testing can be isolated to a single component: Oracle Internet Directory.

LDAP-Enabled Oracle Products

The following Oracle9i products use Oracle Internet Directory:

- **Oracle Net Services**

Oracle Net Services encompasses features that provide database access control, network connectivity, manageability, and scalability. Oracle Net, an Oracle Net Services component, uses Oracle Internet Directory as a primary method for storing and resolving database connect identifiers.

- **Oracle Advanced Security**

Oracle Advanced Security provides a number of features that protect enterprise networks. These features encompass encryption, authentication, single sign-on, and security protocols. Oracle Advanced Security uses Oracle Internet Directory as a central repository for user authentication and authorization information.

- **Application Context**

Application Context is a database security feature that enables you to base applications on a user's session information. A centrally initialized application context uses Oracle Internet Directory to store the context's values.

- **Oracle Advanced Queuing**

Oracle Advanced Queuing is a feature that enables distributed applications to send messages to one another asynchronously. Oracle Advanced Queuing uses Oracle Internet Directory to store metadata for global topics and registrations.

- **Oracle Dynamic Services**

Oracle Dynamic Services provides e-businesses with a method for registering and reusing Internet, Intranet, and database information services. Oracle Dynamic Services uses the directory to store service definitions and application profiles.

Oracle Internet Directory

Oracle Internet Directory is Oracle's directory service compliant with LDAP version 3. It runs as an application on the Oracle9i database, which may or may not reside on the same operating system. To communicate with the database, Oracle Internet Directory uses Oracle Net Services, remote data-access software that enables client-to-server and server-to-server communication across any network.

Oracle Internet Directory's scalability, high availability, and security features make it the directory of choice for enterprise applications.

- **Scalability**

Because it runs on powerful Oracle9i, Oracle Internet Directory can store terabytes of information. At the same time, multithreading and database connection pooling enable it to handle thousands of concurrent users and achieve subsecond search response times.

- **High Availability**

Oracle Internet Directory supports all Oracle 9i high-availability solutions and technologies, such as clustered "logical hosts," Real Application Clusters, failover, and multimaster replication. These solutions ensure that, if one server fails, a user can access the most current information from another server.

- **Security**

Oracle Internet Directory has comprehensive and flexible security features. The security administrator can confine access to specific directory objects or expand it to entire directory subtrees. Three levels of security are possible: anonymous, password-based, and certificate-based using Secure Sockets Layer version 3.

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

Integrating Oracle Products with Third-Party Directories

Oracle Internet Directory includes the Oracle Directory Integration Platform. This platform synchronizes data between Oracle Internet Directory and different directories within an organization. These directories might include NOS directories, groupware address books, applications such as HR, and metadirectories.

Metadirectories consolidate disparate information by propagating changes to the different directories that an organization contains. The Oracle Directory Integration Platform enables customers to build a single directory with a global directory entry that contains information from multiple sources.

Oracle Directory Integration Platform consists of the following components:

- Directory Integration Agents, which provide connectivity between Oracle Internet Directory, other Oracle products, and third-party directories.
- Directory Integration Server, which controls the scheduling and running of the agents.
- Directory Integration Toolkit, which allows third-party metadirectory vendors to develop agents and to connect their metadirectory solutions with Oracle Internet Directory.

Directory Server Concepts

This chapter lays a foundation for understanding LDAP-compliant directories. It begins by describing the function of directories—be they paper based or electronic—proceeds through a definition of the LDAP protocol, version 3, and ends by identifying the fundamental components of an online directory.

The chapter covers the following topics:

- [What Is a Directory?](#)
- [What Is an Online Directory?](#)
- [Directories and Relational Databases Compared](#)
- [Typical Directory Applications](#)
- [The Benefits of Standards-Based Online Directories](#)
- [What Is LDAP?](#)
- [Directory Information Flow: An Example](#)
- [Directory Components and Features](#)
- [The Oracle Context](#)

What Is a Directory?

A directory is an index or list that helps people find information. The directories most familiar to us are offline, usually paper based, resources like telephone books and yellow pages, merchandise catalogs, card catalogs in libraries, and dictionaries.

What Is an Online Directory?

Online directories are computer databases that serve much the same function that offline directories do, but add the following benefits.

- **Flexibility**
Online directories can organize data in many different ways, allowing users to specify different search criteria.
- **Security**
Online directories centralize data, making it easier to manage and to restrict access to it.
Dynamism
Online directories can be updated frequently.
- **Personalization**
Online directories enable you to globally store user profiles, the color settings on your personal computer, for instance.

These added benefits make online directories ideal for storing critical information at large companies. Typical entries in an online directory include employee names, enterprise roles, e-mail addresses, and information about printers, conference rooms and other company resources.

Directories and Relational Databases Compared

It is easy to confuse a directory with a relational database, because a directory is, after all, a special kind of database. But there are some significant differences between the two, as [Table 2-1](#) on page 2-3 shows.

Table 2–1 Directories Versus Relational Databases

Directories	Relational Databases
Read more frequently than written	Written more frequently than read
Handle small, simple units of data	Handle large, complex, transaction-oriented units of data
Distributed widely	Not distributed widely
Store information in hierarchically arranged entries	Store information as records in relational tables

Read-to-Write Ratio

A directory is sometimes read 1,000 to 10,000 times more than it is written. That is because it stores information that is updated infrequently, but accessed constantly—information such as user IDs, e-mail addresses, and catalog data. Relational databases by contrast serve as repositories for data that changes frequently, such sales orders, salaries, and student grades. As a result, they are written to frequently, but read infrequently.

Data Units

Directory objects are generally small because they must be representable in an attribute format—for example, `surname=hay`. This feature optimizes the directory for searching. Databases by contrast can accommodate large objects.

Distribution

Directory applications expect at all times to see the same information throughout the deployment environment—regardless of which server they are querying. If a queried server does not store the information locally, then it must either retrieve the information or point the client application to it transparently. A relational database, while it can be distributed, usually resides on a particular server.

Entries

Just as the small size of directory objects optimizes the directory for searching, so too does the way the objects are stored. Represented as a discrete entry in a directory information tree, each piece of directory data can be retrieved quickly. A database search operation, on the other hand, is more suitable for relational transactions—that is, transactions that encompass several pieces of data and several tables.

Typical Directory Applications

Common directory applications include the following:

- Online telephone books
These might serve as a repository not only for phone numbers, but for e-mail addresses and employee names.
- E-mail applications
E-mail servers, for instance, require access to e-mail addresses, user names, mailbox locations, and routing and protocol information. These data categories are all suitable for directory storage.
- HR applications
These require detailed information about people, information that is easily stored in a directory. This information consists of employee identification numbers, birth dates, salary levels, hire dates, and job titles.

The Benefits of Standards-Based Online Directories

The primary benefit of online directories is that they can centralize the storage of information. This feature is critical in a distributed database environment, and it cannot be accomplished without a common standard that governs how enterprise applications interact with directories. Without such a standard, large companies might have to deploy hundreds of application-specific directories, all equipped with their own protocols. Collectively, these application-specific directories pose three major problems:

- Inconsistent data: Information that is updated in one directory might not be updated in others.
- Data redundancy: Entries must be duplicated across directories.

- **Administrative headaches:** Application-specific directories increase the time and cost of managing them because directory entries must be entered or modified not once, but many times.

These problems become apparent when, for example, an employee leaves a company or transfers to another department. When this happens, network administrators might have to disable multiple accounts on multiple databases. The time required to effect these changes and the difficulty involved in synchronizing them across databases is an administrative burden and also a security risk.

Fortunately, Lightweight Directory Access Protocol (LDAP) eases the burden of managing application-specific directories.

What Is LDAP?

LDAP is a standard, extensible directory access protocol that enables directory clients and servers to interact using a common language. LDAP, as the name suggests, is a lightweight implementation the X.500 Directory Access Protocol (DAP), first published in 1990. The X.500 protocol grew out of a need for a directory model that bridged applications and operating systems. However, it proved cumbersome, partly because it runs over the OSI networking stack. LDAP by contrast runs directly over TCP/IP, which is popular, fast, simple, and relatively inexpensive to implement.

This section contains the following topics:

- [The Benefits of LDAP](#)
- [LDAP Version 3](#)
- [C LDAP API](#)
- [LDIF](#)

The Benefits of LDAP

LDAP simplifies directory management in the following ways.

- It provides users and applications in an enterprise with a single, well-defined, standard interface to a single, extensible directory service.
- It reduces the need to manage and coordinate application-specific directories
- Its well-defined protocol and array of programmatic interfaces make it more practical to deploy internet-ready applications that leverage the directory.

LDAP Version 3

The most recent version of the LDAP protocol is version 3, which in December 1997 was approved as an Internet standard. Version 3 improves on version 2 in five ways.

- Globalization Support

LDAP version 3 supports UTF-8, an encoding of Unicode, the 16-bit encoding standard used to store and retrieve information in any language.

- Referrals

LDAP 3 supports knowledge references, LDAP URLs that refer users to other directory servers if the requested information does not reside on the server being queried. This feature enables a directory to be partitioned—that is, distributed across different servers.

- Security

LDAP version 3 supports SASL (Simple Authentication and Security Layer), an Internet standard that enables clients to choose the authentication protocols that they want to use. It also supports Transport Layer Security (TLS), a standardized version of Secure Sockets Layer (SSL), which encrypts data that passes between client and server.

- Extensibility

LDAP version 3 enables new LDAP operations to be defined, uses mechanisms called controls to modify existing operations, and permits new authentication methods through SASL.

- Feature and schema discovery

LDAP version 3 servers publish the versions of the LDAP protocol that they support and their schemas in a directory entry called the root DSE (directory server-specific entry). This feature facilitates interaction with other LDAP clients and servers.

See Also: RFCs (Request for Comments) 2251–2256 on the IETF Web site at <http://www.ietf.org/>

C LDAP API

The C LDAP API, introduced with LDAP version 2, provides a standard API for accessing and modifying directory entries from the command line. The API offers the programmer of an LDAP-enabled application a set of functions that covers every LDAP protocol operation.

APIs for the Java and Perl programming languages are also available.

See Also:

- RFC (Request for Comment) 1823 on the IETF Web site at <http://www.ietf.org/>. This RFC documents the LDAP C API for LDAP
- Appendix B, "LDAP Command-Line Tools"

LDIF

Lightweight Directory Interchange Format (LDIF) is a text-based format used to describe and modify—change, add, and delete—directory entries. In the latter capacity, it provides input to command-line utilities.

The two LDIF files immediately following represent a directory entry for a printer. The string in the first line of each entry is the entry's name, called a distinguished name. The difference between the files is that the first describes the entry—that is, the format is an index of the information that the entry contains. The second, when used as input to the command-line utility `ldapmodify`, adds information about the speed of the printer.

Description

```
dn: cn=LaserPrinter1, ou=Devices, dc=acme,dc=com
objectclass: top
objectclass: printer
objectclass: eponPrinter
cn: LaserPrinter1
resolution: 600
description: In room 407
```

Modification

```
dn: cn=LaserPrinter1, ou=Devices, dc=acme, dc=com
changetype: modify
add: pagesPerMinute
pagesPerMinute: 6
```

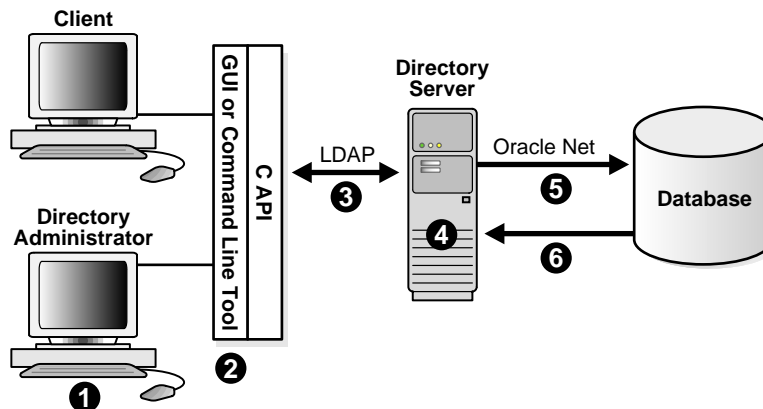
Directory Information Flow: An Example

To visualize how information is retrieved from an LDAP-compliant directory, consider how the process works in Oracle Internet Directory:

1. The client issues a search request, using either a graphical user interface (GUI) tool or a command-line tool and one or more authentication methods.
2. The command-line tool or the GUI tool invokes the C API, directly if the command-line tool is used and by way of the Java Native Interface if the GUI tool is used.
3. The search request is transmitted to the directory server using the LDAP protocol.
4. The directory server authenticates, or binds, the client and then checks access control lists (ACLs) to determine whether it can grant the client's request.
5. The directory server transmits the request to the database server, using the remote database access software Oracle Net to convert the search request from LDAP to language that the database can understand.
6. The database retrieves the requested information, sending it back to the directory server, back to the C API, and back to the client.

Figure 2-1 on page 2-8 illustrates the process.

Figure 2-1 Information Flow for Oracle Internet Directory



Directory Components and Features

This section describes the information that a directory contains. It explains how this information is organized and who gains access to it. The section contains the following topics:

- [Entries](#)
- [Attributes](#)
- [Types of Attributes](#)
- [Attribute Syntaxes and Matching Rules](#)
- [Foreign-Language Options for Attributes](#)
- [Object Classes](#)
- [Types of Object Classes](#)
- [Creating New Object Classes and Redefining Old Ones](#)
- [Naming Contexts](#)
- [Schema](#)
- [Security](#)

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 a printer or other shared resource, a department within a company, or even the company itself.

To name it and to identify its location in the directory hierarchy, each entry is assigned a unique distinguished name (DN). The DN of an entry consists of the entry itself, known as the relative distinguished name (RDN), and its parent entries, connected in ascending order, from the entry itself up to the root (top) entry in the tree. Collectively, these entries form a directory information tree (DIT) such as the one shown in [Figure 2-2](#) on page 2-10. A directory server uses this tree to determine what information to extract from a relational, or other, database.

Figure 2-2 Directory Information Tree with a Distinguished Name Highlighted

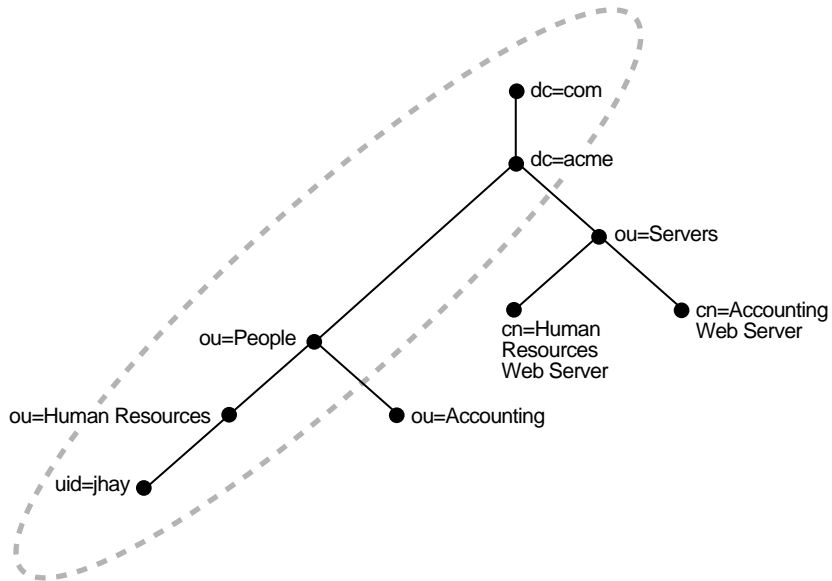


Figure 2-2 represents a portion of a DIT belonging to the company acme, designated by the entry dc (domain component) =acme, dc=com. The highlighted DN uid=jhay, ou=Human Resources, ou=People, dc=acme, dc=com is an entry within the DIT.

Note: Spaces after commas are optional for DNs.

This entry represents the user ID (uid) for a person belonging to the organizational unit (ou) Human Resources in the company acme.

The format of a DN places the lowest hierarchical component of the name, the RDN, to the extreme left. In the example, this RDN is uid=jhay.

Attributes

An entry consists of a set of attributes, each describing a unique feature of the entry. An attribute consists of two components, an attribute type and one, or sometimes more, values. [Table 2–2](#) lists, in LDIF notation, some of the attributes that the entry `uid=jhay, ou=Human Resources, ou=People, dc=acme, dc=com` might contain.

Table 2–2 A Typical Directory Entry with Attributes Shown

Attribute Type	Attribute Value
<code>cn:</code>	John Hay
<code>cn:</code>	Jack Hay
<code>givenname:</code>	John
<code>sn:</code>	Hay
<code>uid:</code>	jhay
<code>mail:</code>	jhay@acme.com
<code>telephoneNumber:</code>	+1 650 555 0167

Note that LDAP permits some of the attributes listed to be abbreviated. The attribute `cn` could just as easily have been written as `commonName`, and the attribute `sn` as `surname`.

Types of Attributes

Attributes take two forms: user and operational. The former are application specific and can be retrieved and modified by the user. The latter are used to control directory operations and are generally not available to the user. Examples of user attributes include `commonName`, `surname`, `telephoneNumber`, and `mail`. Examples of operational attributes include the following:

- `modifyTimeStamp`—the date and time an entry was last changed
- `modifiersName`—the DN that last made the change
- `supportedLDAPVersion`—the LDAP versions supported by the directory server.

Attribute Syntaxes and Matching Rules

Under LDAP rules, each attribute type must conform to a particular syntax and associated matching rule. A syntax determines the form that an attribute value takes. A matching rule specifies how attribute values are compared in directory searches.

[Table 2–3](#) lists common syntaxes and associated matching rules defined by the X.500 standard.

Table 2–3 *Attribute Syntaxes and Associated Matching Rules*

Syntax	Matching Rule
DirectoryString	caseIgnoreMatch
Text string	Ignore letter case and leading, trailing, and multiple spaces caseExactMatch Match case of letters. Ignore letter case and leading, trailing, and multiple spaces
PrintableString	telephoneNumberMatch
Text string for a telephone number	The same as caseIgnoreMatch, but it also ignores space and hyphen characters
Integer	integerMatch
Numbers	Follow rules for comparing integers
DistinguishedName	distinguishedNameMatch
Directory names	Follow special rules for comparing DNs
OctetString	octetStringMatch
Binary data	Compare data byte by byte

Note: Your directory might use different names for syntaxes and matching rules depending upon the kind of schema format that it uses to describe entries.

Suppose that you are searching the directory for an employee named Kit Karston. Following X.500 rules, the syntax used to represent this name is `DirectoryString`. The matching rule can be either `caseIgnoreMatch` or `caseExactMatch`. If it is the former, you might enter the name as `(cn=kit karston)` or `(cn=kitKarston)` or even `(cn= kit karston)`. In all cases, the directory returns the name.

Foreign-Language Options for Attributes

In addition to containing multiple values, attributes can store language codes. This feature is useful for accessing text in the many languages that LDAP supports. For example, the attribute `cn;lang-ja` represents a common name in Japanese. Note that a semicolon separates the attribute type and the value.

You can also specify that the directory return attributes in a given dialect. For example, the language code `lang-en-GB`, returns attribute values in British English.

Object Classes

An object class is a collection of attributes that you use to define an entry. Some of these attributes are mandatory; others are optional.

If, for example, you assign the LDAP-defined object class `organizationalPerson` to the entry `uid=jhay`, `ou=Human Resources`, `ou=People`, `dc=acme`, `dc=com`, you must include `commonName (cn)` and `surname (sn)` as attributes for the entry. Rules for the object class `organizationalPerson` also allow you to include the attributes `telephoneNumber`, `uid`, and `userPassword`, but these are not required.

Excluding optional attributes, the entry preceding might look something like this in LDIF notation:

```
dn: uid=jhay, ou=Human Resources, ou=People, dc=acme, dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: John Hay
cn: Jack Hay
sn: Hay
```

Note that three object classes are present in the entry, an indication that object subclasses are represented. In this case, `organizationalPerson` is a subclass of the object class `person`, which is a subclass of the object class, `top`.

In addition to defining the attributes of an entry, object classes provide a way of locating a related group of entries. To restrict your directory search to printers housed in a certain area of your organization, for instance, your directory access GUI might construct an LDAP search filter that uses an AND operator to combine the object class `printer` with the attribute `description`, which might contain a value for the location of the printers.

See Also: ["Creating New Object Classes and Redefining Old Ones"](#) on page 2-15, for a discussion of object subclasses

Types of Object Classes

Object classes take three forms:

- [Structural Object Classes](#)
- [Auxiliary Object Classes](#)
- [Abstract Object Classes](#)

Structural Object Classes

Most of the object classes in a directory are structural, because they define what an entry is. They also impose rules on the entries that are stored beneath them. For example, the object class `organization (o)` might require that all objects stored beneath it belong to the object class `organizational units (ou)`. Other examples of structural object classes are `person`, `printer`, and `groupOfNames`.

Auxiliary Object Classes

LDAP rules require each entry to belong to one, and only one, structural class, but an entry can also belong to one or more auxiliary classes. An auxiliary class, as its name suggests, is used to add attributes to entries that are already defined by a structural object class. Note that an auxiliary class cannot stand on its own in an entry. The entry must also contain a structural object class. Unlike structural object classes, auxiliary classes place no restrictions on where an entry is stored.

Abstract Object Classes

The third type of object class, abstract, is a class whose primary function is to determine the structure of an LDAP directory. The object class `top`, for example, is the root object class from which all structural object classes are derived. It contains one mandatory attribute, `objectClass`, and because all entries inherit its attributes, it ensures that these entries are defined by an object class. An abstract object class cannot stand alone in an entry. The entry must also contain a structural object class.

Note: Some directory vendors may not distinguish between object class types and therefore may not enforce structure rules.

Creating New Object Classes and Redefining Old Ones

LDAP enables you to create entirely new structural object classes and attributes to accommodate new objects in a directory. The challenge in creating new object classes is to come up with unique names because object class and attribute namespaces are flat.

A far more common—and easier—practice is to create subclasses of existing classes, which then become superclasses. This feature provides a way of adding mandatory, as well as optional attributes to a predefined object class. A subclass inherits all of the attributes of a superclass, and because an entry can contain more than one object class, it can inherit numerous attributes.

The object class `printer`, for example, might have the object class `epsonPrinter` as a subclass to provide information about a specific kind of printer that an organization uses.

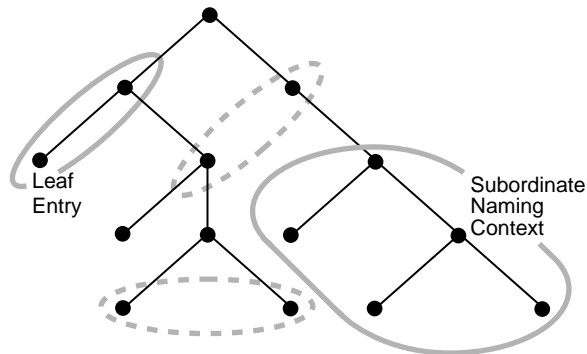
Auxiliary object classes provide the easiest, most flexible method for redefining existing directory entries because you are not required to subclass them to specific object classes and can use them to add attributes to any number of entries. An object class that can be used to add uniform resource locators (URLs) to any directory entry is a good example of an auxiliary object class.

Naming Contexts

A naming context is a DIT that resides entirely on one server. It can consist of a single entry, a subtree, or even the entire DIT. Any directory entry can serve as the root of a naming context as long as the entries below it are contiguous with it. These subordinate entries can be either leaf entries or naming contexts in their own right.

Some, but not all, of the naming and non-naming contexts in [Figure 2–3](#) are highlighted. Naming contexts are indicated by solid-line circles, non-naming contexts by dotted-line circles.

Figure 2–3 Naming Contexts and Non-Naming Contexts



To enable you to specify specific naming contexts as search targets, LDAP enables you to publish them in a directory's root DSE (directory server-specific entry). You publish a naming context by assigning the root, or topmost entry, of the context as a value to an attribute called `namingContexts`.

Schema

The schema of a directory comprises the metadata that determine what objects a directory can store. The metadata of a directory are its object classes, attribute types, attribute syntaxes, and matching rules.

The schema of a directory typically amounts to dozens of object classes, hundreds of attributes, and a dozen or fewer syntaxes.

To facilitate directory navigation and modification, LDAP version 3 requires directories to publish their schema in an operational attribute called `subschemaSubentry`, located in a directory server's root DSE (directory server-specific entry). This attribute is analogous to the data dictionary of a relational database. It is in the entry `subschemaSubentry` that you add new object classes and attributes and redefine existing object classes.

See Also: [Appendix A, "Oracle-Specific LDAP Schema Extensions"](#)

Security

Gaining access to a directory is a two-part process that consists of using one or more authentication methods to establish the identity of a directory client and then using access control lists (ACLs) to determine what kind of information clients can access and what they can do with it once they have accessed it.

Authentication

LDAP version 3 supports four levels of authentication:

- **Anonymous**
Users log in without providing a user name and password, but once they are logged on, they may have limited privileges.
- **Simple**
Clients supply a user name, in the form of a DN, and an unprotected password—that is, a password sent in clear text.
- **Simple over SSL**
Users supply a user name and a password, which is protected using Secure Sockets Layer (SSL), a public key encryption technology.
- **SSL with certificates**
This method provides maximum protection because it supplements public key encryption with certificates that clients use to authenticate themselves. Because they are issued by a certificate authority, certificates provide a considerable degree of certainty about a client's identity.

Access Control Lists

Once you gain access to a directory, a mechanism called an access control list (ACL) determines what kind of information you are able to retrieve and modify.

An ACL consists of one or more operational attributes called ACIs (access control items). These ACIs specify permissions for an entry. Theoretically, you can place an ACL anywhere in the directory hierarchy, down to the level of an entry. In reality, ACL placement is subject to whatever restrictions your directory software imposes. An ACL specifies three things:

- The directory objects that are subject to access control
- The clients that are granted or denied access

- The access rights that clients are granted

The following example shows the format of an ACL that is constructed using the command line tool `ldapmodify`. This ACL is based on the Oracle Internet Directory attribute `orclEntryLevelACI`, which sets access control rules for one entry only.

```
dn: uid=jhay, ou=Human Resources, ou=People, dc=acme, dc=com
changetype: modify
replace: orclentrylevelaci:
orclentrylevelaci: access to entry
  by dn= "cn=directory manager, dc=acme, dc=com" (browse, add, delete)
  by * (browse, noadd, nodelete)
orclentrylevelaci: access to attr=(*)
  by dn= "cn=directory manager, dc=acme, dc=com" (search, read, write,
compare)
  by * (search, read, nowrite, nocompare)
```

The ACL above consists of two ACIs (in boldface) that set access control rules for the entry `uid=jhay, ou=Human Resources, ou=People, dc=acme, dc=com`. These ACIs give directory managers falling within the domain `dc=acme, dc=com` read and modify privileges over the entry and its attributes. They give all other users, designated by the wildcard "*" read but not write privileges. The entity assigned privileges by an ACI can be a privilege group as well as an individual.

The Oracle Context

A directory stores all information pertaining to Oracle software under one or more entries called an Oracle Context, which has an RDN of `cn=OracleContext`. You can create an Oracle Context under any entry in the DIT. To help you, Oracle Net Configuration Assistant, an Oracle tool for configuring directory access, displays a list of published entries as suggested locations. If you are using Oracle Internet Directory, a starter Oracle Context is created when the directory is installed.

A starter Oracle Context, as [Figure 2-4](#) on page 2-20 illustrates, is a directory subtree that, at the top level, consists of four containers—`Products`, `Groups`, `Services`, and `Computers`—and entries for three of the administrative groups applicable to the entire context. The only product-related entries installed at this stage are entries for enterprise user security and Enterprise Security Manager, a GUI tool.

After using Oracle Net Configuration Assistant to configure access to a directory, you can use another tool, Database Configuration Assistant, to register databases. Registration adds entries for database servers and their associated Oracle Net connect descriptors.

Figure 2-4 Starter Oracle Context with Added Entries for a Database and a Database Connect Descriptor

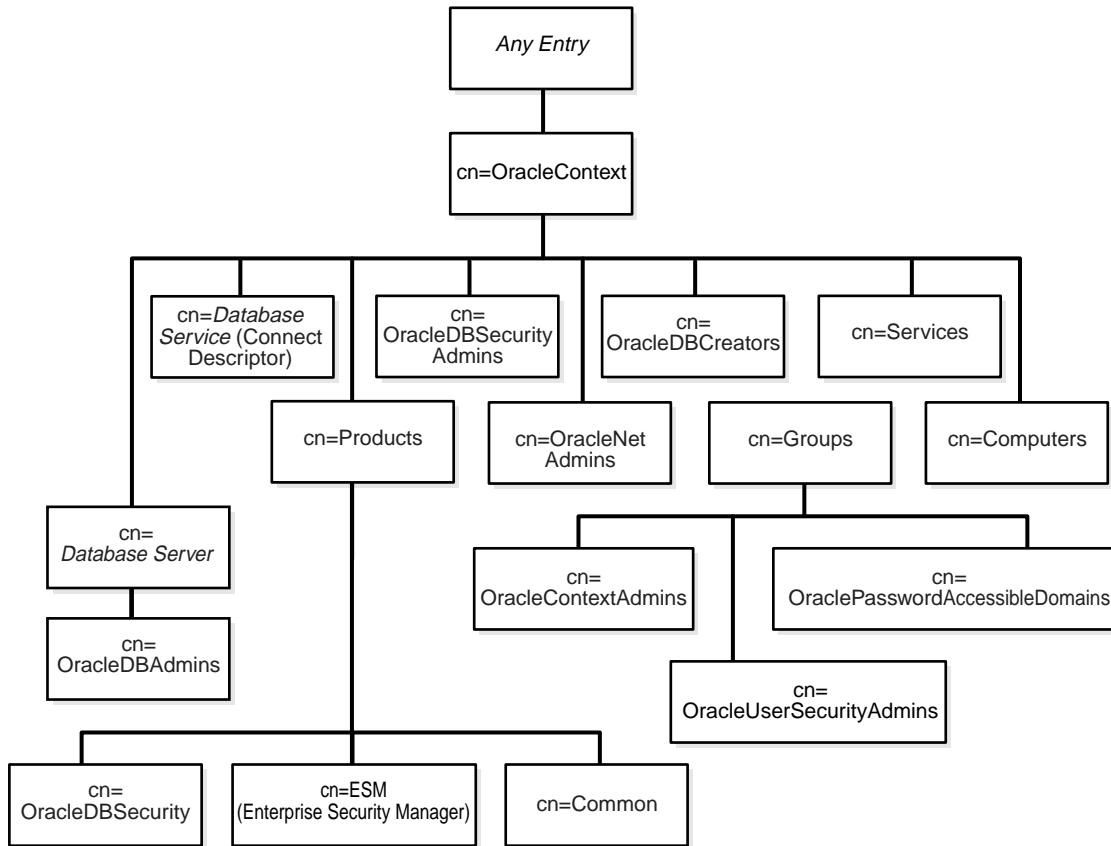


Table 2-4 shows the contents of the four containers represented in Figure 2-4.

Table 2-4 Containers Under the Oracle Context

Container	Description
Products	The <code>Products</code> container is the repository for product-specific entries for every Oracle product except Oracle Net Services and Oracle Advanced Queuing. The entries in this container are private to the product and may be protected by access control policies. The entry <code>Common</code> (<code>cn=Common</code>) stores attributes common to all objects—for instance, attributes that uniquely identify a user.
Groups	The <code>Groups</code> container stores entries for the administrative groups applicable to the entire Oracle Context. At present, these groups are <code>OracleContextAdmins</code> , <code>OracleUserSecurityAdmins</code> , and <code>OraclePasswordAccessibleDomains</code> . Entries for the groups <code>OracleNetAdmins</code> , <code>OracleDBCreators</code> , and <code>OracleDBSecurityAdmins</code> appear just below the Oracle Context.
Services	The <code>Services</code> container stores entries for the services offered by Oracle products. Storing these entries separately facilitates discovery of required services. In the future, <code>Services</code> will be a repository for database server entries.
Computers	The <code>Computers</code> container stores entries containing information about each machine—for example, configuration information for a particular server running in a particular Oracle home on a particular machine.

See Also:

- Howes, Timothy A., Mark C. Smith, and Gordon S. Good, *Understanding and Deploying LDAP Directory Services*, Macmillan Technical Publishing, 1999
- *Oracle Internet Directory Administrator's Guide*

Planning and Deployment Guidelines

This chapter provides an overview of the issues you should consider before deploying a directory. For detailed information about how to deploy Oracle Internet Directory, see Part VI of *Oracle Internet Directory Administrator's Guide*.

The chapter covers the following topics:

- [What Do You Put in a Directory?](#)
- [Designing an Effective Directory Tree and Choosing Entry Names](#)
- [Physically Distributing the Directory: Partitions and Replicas](#)
- [Designing for High Availability and Failover](#)
- [Capacity Planning, Sizing, and Tuning](#)
- [Designing Directory Security](#)

What Do You Put in a Directory?

The most important factor to consider when trying to decide what to store in a directory is that a directory is no substitute for a database. Because directories are designed for read operations, you should avoid using them as repositories for information that will change often. By keeping write operations in a directory to a minimum, you improve search performance. Directories are fine for single operations involving one directory entry, operations that involve relatively static information such as administrative metadata. But for transactional operations involving multiple data items and more than one operation, databases are the preferred repository.

The following are suitable candidates for directory storage:

- Contact information such as phone numbers, e-mail addresses, and geographical addresses
- Employee profiles that consist of information about a worker's salary, job title, manager, and department
- Software configuration information
- Software preferences
- Nontransactional billing information such as credit limits, credit card numbers, customer contact information
- Pointers to large objects, which are not suitable for directory storage, objects such as JPEG images and Java applications

Designing an Effective Directory Tree and Choosing Entry Names

Designing an effective directory information tree and assigning effective names for entries requires careful planning and enterprise-wide coordination. An effective directory structure incorporates the following features:

- It borrows rules for assigning names and numbers to employees from a company's Human Resources Department, whose policies are valid throughout the enterprise. The alternative is to come up with unique names, a process that adds administrative overhead.
- It avoids organizing entries by corporate hierarchy, preferring instead to include a person's organizational information as an attribute in his or her directory entry. By not using the corporate hierarchy as the measure of how the directory is organized, you avoid the administrative disruptions that frequent corporate reorganizations cause.

- It organizes the directory information tree to reflect data ownership boundaries. This practice makes it easier to develop effective access control and replication policies. For example, a multinational corporation that wants to consolidate its global directory can accomplish this goal by dividing the directory into naming contexts that correspond with geographic regions, each subject to its own access control and replication policies.

Physically Distributing the Directory: Partitions and Replicas

The model of a centralized, consolidated directory, and the cost savings associated, cannot be achieved without multimaster replication. Using this technology, two or more directory nodes in a network store a copy of the directory and each of them updates the directory and replicates the changes to the other nodes. Because replication can occur at the level of a naming context, the organization can avoid the administrative burden associated with partitioning the directory across different servers.

A strong, centralized directory has the following features:

- It consists of a network of two or more directory nodes, each containing all the naming contexts, all bound by a multimaster configuration.
- Its nodes are deployed, one in each geographic region, to suit the corporate data network connectivity. For example, if a region is connected to the rest of the network by way of a slow link, it is better to locate a dedicated directory server for use by the clients in that region.
- Its regional servers are each configured for failover and recovery.

Reasons to Replicate

Directory replication is desirable under the following circumstances:

- The organization consists of widely dispersed data centers that require a common directory but are interconnected with low bandwidth links involving multiple intermediate routers.
- The number of clients accessing the directory server exceeds server capacity, and load balancing is required.
- The organization wants to ensure system availability in the event that the directory server fails.

See Also: Chapter 14, "Managing Directory Replication" in *Oracle Internet Directory Administrator's Guide*

Reasons to Partition

Partitioning a directory over two or more servers is expensive because each partition must have its own plan for backup, recovery, and other data management functions. Unless the partitions of your directory are characterized by the following conditions, you should plan to replicate it.

- The partition corresponds to administrative and data ownership boundaries that are better left independent.
- The organization consists of widely dispersed data centers that are interconnected with low bandwidth links but have only local access needs.
- The partition is not integral to the organization at large.
- The expense of maintaining replicas of entire directories is insupportable.

Designing for High Availability and Failover

Multimaster replication ensures that a directory is always available, and it provides a failover remedy, but you should also be aware of two other backup and recovery methods, Intelligent Client Failover and Intelligent Network Level Failover. Both of these are options where Oracle Internet Directory is installed.

Intelligent Client Failover enables clients connecting to Oracle Internet Directory to contact alternate server instances of Oracle Internet Directory if their connection to a given server instance fails.

Intelligent Network Level Failover is a technology that detects failure in the server hosting Oracle Internet Directory and reroutes connection requests to other servers. It has load balancing and failover capabilities.

See Also: Chapter 19, "Managing High Availability and Failover," in *Oracle Internet Directory Administrator's Guide*

Capacity Planning, Sizing, and Tuning

Determining the load and capacity requirements of any given directory node requires foresight and careful planning. It consists of three discrete processes: capacity planning, sizing, and tuning.

This section contains the following topics:

- [Capacity Planning](#)
- [Sizing](#)
- [Tuning](#)

Capacity Planning

Capacity planning involves determining the load that a directory server will bear and the capacity it must have. These are a function of the following factors:

- The type of LDAP client applications accessing the server
- The number of users accessing these applications
- The kind of LDAP operations that these applications perform
- The number of entries in the directory information tree
- The type of operations the directory server performs
- The number of concurrent connections to the directory server
- The peak rate at which the directory server must perform operations
- The average latency tolerable under peak load conditions

See Also: Chapter 17, "Capacity Planning," in *Oracle Internet Directory Administrator's Guide*

Sizing

Once you have determined the load and capacity requirements of a directory server, you can determine system requirements. Pay attention to the following factors:

- The type and number of CPUs for the directory server computer
- The type and size of disk subsystems for the directory server computer
- The amount of memory required by the directory server computer
- The type of network used for LDAP messages from clients

See Also: Chapter 17, "Capacity Planning," in *Oracle Internet Directory Administrator's Guide*

Tuning

Before actually using your directory, you should test it, using as test data the applications that will interact with it. Any tool that you devise for testing should use overall throughput and the average latency of operations as a measure of how tuning should be performed.

The more commonly tuned properties are:

- CPU usage

This is contingent on the number of directory servers deployed and the number of database connections that each server opens.

- Memory usage

In the case of Oracle Internet Directory, the biggest consumer of memory is the database cache. It should be tuned so that physical memory is always available. Too large a cache causes paging, which impedes performance. Too small a cache causes excessive disk I/O, which also impedes performance.

- Disk usage

If the data returned by the directory resides in database tablespaces, you can do the following to improve data throughput:

- Balance tablespaces on different logical and physical drives
- Stripe logical volumes onto multiple physical volumes
- Distribute disk volumes across several I/O controllers

See Also: Chapter 18, "Tuning," in *Oracle Internet Directory Administrator's Guide*

Designing Directory Security

When designing directory security, do the following:

- Grant the least amount of access that you can.
- Think carefully about the kind of access control lists (ACLs) that you place at the root of the directory tree because such placement determines the kind of access that users have to the rest of the tree.
- Use groups instead of individuals in ACLs as much as possible.
- Note that group permissions at a higher level of the directory tree cannot currently be reversed at a lower level.

See Also: Chapter 12, "Managing Directory Access Control,"
in *Oracle Internet Directory Administrator's Guide*

Deploying Oracle Products with Oracle Internet Directory

This chapter takes an in-depth look at how Oracle9i products interact with Oracle Internet Directory. It describes how each product uses the directory, where under the Oracle Context the product stores its entries, and how the product protects these objects from unauthorized access. Where appropriate, the chapter talks about deployment factors that you should be aware of before using Oracle Internet Directory.

The chapter covers the following products:

- [Oracle Net Services](#)
- [Oracle Advanced Security](#)
- [Application Context](#)
- [Oracle Advanced Queuing](#)
- [Oracle Dynamic Services](#)

Oracle Net Services

Oracle Net Services provides enterprise-wide connectivity solutions in distributed, heterogeneous computing environments. Oracle Net Services eases the complexities of network configuration and management, maximizes performance, and improves network diagnostic capabilities. It provides the following solutions for a typical network configuration:

- **Connectivity**

Once a network session is established, Oracle Net, a component of Oracle Net Services, acts as the data courier for the client application and the database server. It is responsible for initiating and maintaining the connection between the client application and database server, as well as exchanging messages between them. Oracle Net is able to perform these jobs because it is located on each computer in the network.

- **Manageability**

Features such as location transparency, centralized configuration, and quick out-of-the-box installation and configuration enable you to easily configure and manage network components.

- **Internet Scalability**

With Oracle Net Services, you can maximize system resources and improve performance. Oracle's shared server architecture increases the scalability of applications and the number of clients simultaneously connected to the database.

- **Internet Security**

Oracle Net Services uses Oracle Advanced Security and other database access control features to enhance network security.

This section covers the following topics:

- [How Oracle Net Services Uses Oracle Internet Directory](#)
- [Oracle Net Services Entries Under the Oracle Context](#)
- [Security Measures for Oracle Net Services Entries](#)
- [Directory Deployment Factors for Oracle Net Services](#)

How Oracle Net Services Uses Oracle Internet Directory

Oracle Net Services uses Oracle Internet Directory as one of the primary methods for storing and resolving connect identifiers to connect descriptors, which are passed back to the client. This feature is called directory naming. A connect identifier is specified in several different ways. One of the most common ways is through the use of a net service name, another name for a database service.

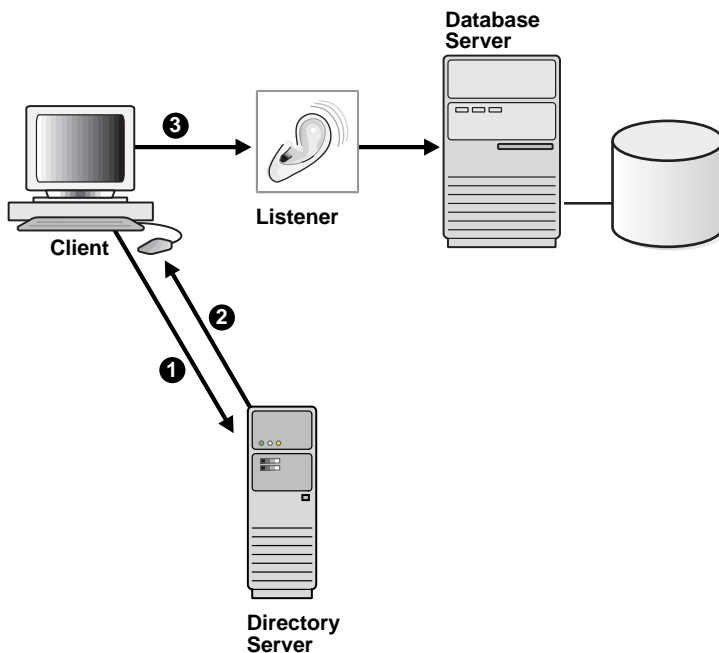
In the following connect string, `sales` is a simple name for a database service that is resolved to connection information that is used to access the database. Instead of storing this information in a `tnsnames.ora` file, you can store it in the directory server.

```
CONNECT username/password@sales
```

[Figure 4-1](#) on page 4-4 shows a client resolving a connect identifier through a directory server.

1. The client contacts the directory server to resolve a connect identifiers to a connect descriptor.
2. The directory server resolves the connect identifier and retrieves the connect descriptor for the client.
3. The client sends the connection request to the listener using the connect descriptor.

Figure 4–1 Client Using a Directory Server to Resolve a Connect Identifier



Note: Java Database Connectivity (JDBC) Drivers support directory naming. See the *Oracle9i JDBC Developer's Guide and Reference* for further information.

Oracle Net Services Entries Under the Oracle Context

As [Figure 4-2](#) shows, directory naming provides support for three kinds of connect identifiers in the directory:

- Database services

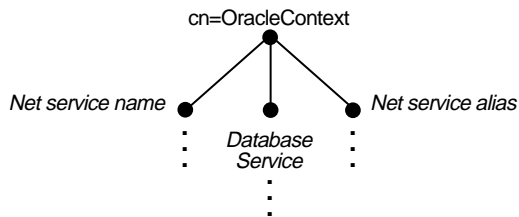
A database service entry contains the actual name of the database, as well as several attributes, including those that constitute the connect descriptor. You create a database service entry when you create the database, using Database Configuration Assistant. The name of the entry matches the database name specified at time of creation. Clients configured to access the directory server can use this entry in their connect strings to connect to the database without any additional configuration.

- Net service names

A net service name is a simple name for a database that resolves to a connect descriptor. A connect descriptor provides the location of the database and the name of the database service. A net service entry contains attributes that constitute the connect descriptor. You create net service name entries with Oracle Net Manager, a graphical user interface tool for configuring and managing Oracle Net Services. The Directory Server Migration Wizard, available within Oracle Net Manager, enables you to export net service names stored in an existing `tnsnames.ora` file to Oracle Internet Directory.

- Net service aliases

A net service alias is an alternative name for a database service or net service name. A net service alias entry does not have connect descriptor information. Instead, it only references the location of the entry 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 is the referenced entry.

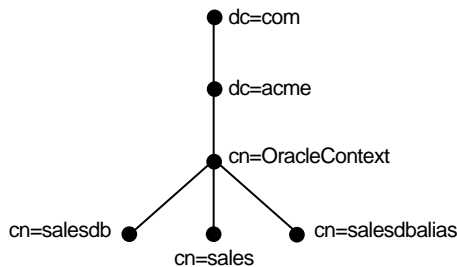
Figure 4–2 Networking Entries

These entries are created directly under the Oracle Context.

In [Figure 4–3](#), the directory contains a database service entry of `salesdb`, a net service name entry of `sales`, and net service alias of `salesdbalias` for `salesdb`. The entries have the following distinguished names (DNs):

- The entry `salesdb` has a DN of
`cn=salesdb,cn=OracleContext,dc=acme,dc=com.`
- The entry `sales` has a DN of
`cn=sales,cn=OracleContext,dc=acme,dc=com.`
- The entry `salesdbalias` has a DN of
`cn=salesdbalias,cn=OracleContext,dc=acme,dc=com.`

When `salesdbalias` is used to connect to a database service, as in `CONNECT username/password@salesdbalias`, it will actually resolve to and use the connect descriptor information for `salesdb`.

Figure 4–3 Example of Networking Entries

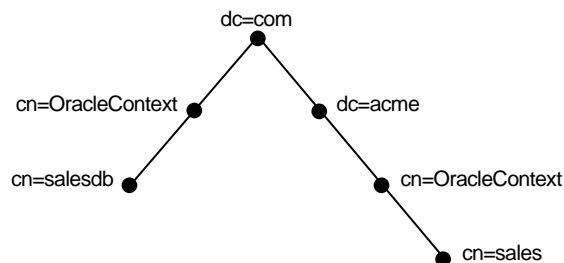
During directory server usage configuration, you select a directory entry that contains an Oracle Context as the default place to locate and look up directory

naming in the directory server. You can use Oracle Net Configuration Assistant to configure directory server usage during or after installation.

If a directory entry lies within the default Oracle Context, you can use a relative path name to gain access to it. In [Figure 4-4](#), the entry `salesdb` has a DN of `cn=salesdb, cn=OracleContext, dc=acme, dc=com` and the entry `sales` has a DN of `cn=sales, cn=OracleContext, dc=acme, dc=com`. If a client needs to access the `sales` entry more frequently than the `salesdb` entry, you would configure `dc=acme, dc=com` as the default directory entry from which to perform lookups. This would enable the client to make an Oracle9i database connection with the following connect string:

```
CONNECT username/password@sales
```

Figure 4-4 Directory Structure with Two Oracle Contexts



In the case where a directory entry that you specify does not lie within the default Oracle Context, you specify the entry's complete name or its absolute name in the client connect string. An absolute name includes the name of the object and its location in the directory server, much the way an absolute path is specified. A client connecting to an Oracle9i database with `salesdb` would use one of the following connect strings:

```
CONNECT username/password@cn=salesdb,cn=OracleContext,dc=com
CONNECT username/password@salesdb.com
```

[Table 4-1](#) lists the object classes for database service, net service name, and net service alias entries.

Table 4-1 Oracle Net Services LDAP Main Object Classes

Object Class	Description
orclDbServer	Defines the attributes for database service entries
orclNetService	Defines the attributes for net service name entries
orclNetServiceAlias	Defines the attributes for net service alias entries

The object classes `orclNetService` and `orclDbServer` use the object classes listed in [Table 4-2](#).

Table 4-2 Oracle Net LDAP Derived Object Classes

Object Class	Description
orclNetAddress	Defines a listener protocol address
orclNetAddressList	Defines a list of addresses
orclNetDescription	Specifies a connect descriptor, containing the listener address for the database and the connect information to the service
orclNetDescriptionList	Defines a list of connect descriptors

These object classes use attributes that specify the contents of connect descriptors.

See Also: Appendix A, "[Oracle-Specific LDAP Schema Extensions](#)"

Security Measures for Oracle Net Services Entries

Oracle Net Services grants read access to the anonymous directory user. This privilege enables any user to access directory naming entries and to use these entries to connect to the database.

While networking entries can be read by anyone, only members of the following groups can create or modify these entries:

- OracleDBCreators group (`cn=OracleDBCreators,cn=OracleContext...`) or the OracleContextAdmins group (`cn=OracleContextAdmins,cn=Groups,cn=OracleContext...`) to create a database service entry with Database Configuration Assistant
- OracleNetAdmins group (`cn=OracleNetAdmins,cn=OracleContext...`) or the OracleContextAdmins group to create net service names or net service aliases with Oracle Net Manager

Oracle Net Configuration Assistant establishes these access rights for the groups during Oracle Context creation.

Directory Deployment Factors for Oracle Net Services

Before deploying directory naming, consider the following:

- You can store directory naming entries under multiple Oracle Contexts.
You can use multiple Oracle Contexts to logically distribute entries by geographic location or other criteria. Oracle Net Configuration Assistant gives clients default access to a specific Oracle Context, but clients can also access entries that lie under other Oracle Contexts.
- You can export data stored in a `tnsnames.ora` file or an Oracle Names server to an Oracle Internet Directory server.

To export net service names stored in a `tnsnames.ora` file, use the Directory Server Migration Wizard, available within Oracle Net Manager. To export database services and net service names stored in an Oracle Names server to the directory server or to an LDIF file, use the Oracle Names Control utility.

Once data is exported, you can either configure clients to use directory naming or, if necessary, convert Oracle Names servers to Oracle Names LDAP Proxy servers to support clients which do not support directory naming. Oracle Names LDAP Proxy servers are Oracle Names servers that have been configured to proxy for directory servers. Upon startup, Oracle Names LDAP Proxy servers obtain network object information from the directory server. This

provides a single point of definition for all data in the directory server and does not require that both Oracle Names server and directory server data be maintained separately and simultaneously.

If you are exporting data from an Oracle Names server with a domain tree to an equivalent directory information tree (DIT) structure, you will have to create an Oracle Context for each subdomain in the DIT.

- You can replicate the domain structure you currently use with Oracle Names, or you can develop an entirely different structure. Introducing an entirely different structure will change the way that clients enter the connect identifier in the connect string. Therefore, Oracle recommends that you consider the relative and absolute naming issues prior to changing the structure.

If you plan to use Oracle Names LDAP Proxy servers that support multiple administrative regions, Oracle recommends mirroring the current Oracle Names structure in the DIT structure. Using a different structure may require modifying the topology defined for the Oracle Names LDAP Proxy servers. The tools for Oracle Net Services do not support topology modification.

- Establish administrative security for directory naming.

Establish administrative privileges for directory naming entries for each Oracle Context. For example, if you want two different sets of administrators to have authority over different directory naming entries, then create two Oracle Contexts.

- Although you must use Oracle Net Configuration Assistant to configure directory usage, you cannot use this tool to create directory naming entries.

Use Oracle Net Manager to create net service name and net service alias entries, and use Database Configuration Assistant to create a database service entry.

See Also: *Oracle9i Net Services Administrator's Guide*

Oracle Advanced Security

Oracle Advanced Security is a term used to describe a number of Oracle features. These features address the administrative and security challenges posed by multiple user accounts on different databases. All rely on the central storage and management of user-related information, such as enterprise roles, in Oracle Internet Directory. For example, when an employee changes jobs, an administrator need only modify information in one location, the directory. This centralization not only lowers administrative costs, it improves enterprise security.

This section covers the following topics:

- [How Oracle Advanced Security Uses Oracle Internet Directory](#)
- [Oracle Advanced Security Entries Under the Oracle Context](#)
- [Security Measures for Oracle Advanced Security Entries](#)
- [Directory Deployment Factors for Oracle Advanced Security](#)

How Oracle Advanced Security Uses Oracle Internet Directory

Oracle Advanced Security uses the directory for the following:

- [Central Management of User Authentication Credentials](#)
- [Central Management of User Authorizations](#)
- [Mappings to Shared Schemas](#)
- [Single Password Authentication](#)
- [Single Sign-On](#)
- [Central Storage of PKI Credentials](#)

Central Management of User Authentication Credentials

A user's database password is stored in the directory as an attribute of his or her user entry, instead of in each database.

Central Management of User Authorizations

Oracle Advanced Security uses directory entries called enterprise roles to determine what privileges a given enterprise user has within a given schema, shared or owned. Enterprise roles are containers for database-specific global roles. User Claire Stevens, for example, might be assigned the enterprise role `clerk`, which might contain the global role `hrclerk` and its attendant privileges on the `human`

`resources` database and the global role `analyst` and its attendant privileges on the `payroll` database.

Mappings to Shared Schemas

Oracle Advanced Security uses mappings, directory entries that point an enterprise user to shared application schema on the database instead of to an individual account. For example, you might map several enterprise users to the schema `sales_application` instead of to separate accounts in their names.

Single Password Authentication

In Oracle *9i*, the Oracle Advanced Security option allows enterprise users to authenticate to multiple databases using a single, centrally managed password. The password is stored in the directory as an attribute of the user's entry and is protected by encryption and access control lists. This feature eliminates the overhead associated with setting up Secure Sockets Layer (SSL) on clients and frees users from having to remember multiple passwords.

Single Sign-On

The alternative to authenticating using a centrally managed password is to use PKI-based single sign-on through SSL. Like single password authentication, this feature requires a user entry in the directory. In addition, a user's wallet must be stored as an attribute of his or her entry.

Central Storage of PKI Credentials

For Oracle *9i*, user wallets can be stored in the directory as an attribute of the user's entry. This feature enables mobile users to retrieve and open their wallets using Enterprise Login Assistant. While the wallet is open, authentication is transparent—that is, users can access any database on which they own or share a schema without having to authenticate again.

Oracle Advanced Security Entries Under the Oracle Context

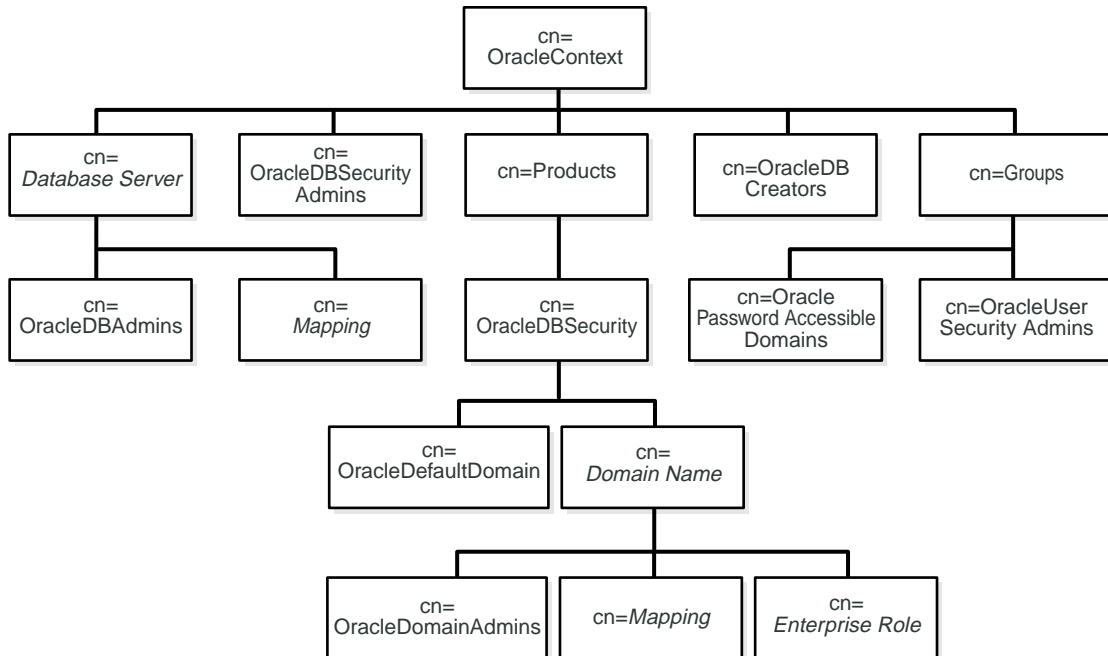
The product subtree for Oracle Advanced Security uses the container `cn=OracleDBSecurity` to store entries for enterprise roles, user-to-schema mappings, and enterprise domains. Under each domain is the entry `cn=OracleDomainAdmins`, which specifies the administrators for the domain.

An enterprise domain is essentially a collection of databases, enterprise roles, and user-to-schema mappings. One of these domains is `cn=OracleDefaultDomain`,

which is created when the Oracle Context is created. This domain can be used in lieu of an administrator-defined domain.

Figure 4-5 shows all entries relevant to Oracle Advanced Security.

Figure 4-5 Directory Entries Relevant to Oracle Advanced Security



Security Measures for Oracle Advanced Security Entries

Oracle Advanced Security uses ACLs at many points in the directory to protect entries relevant to database security. Most of these ACLs grant privileges to members of the groups whose functions are described in [Table 4-3](#).

Table 4-3 Administrative Groups for Oracle Advanced Security

Administrative Group	Function
OracleDBSecurityAdmins	Full privileges for objects in the container <code>OracleDBSecurity</code> . The initial member of this group is the context creator
OracleDomainAdmins	Full privileges for a given domain. The initial member is the person creating or updating the domain. If a new 9i context and <code>OracleDefaultDomain</code> is created, the initial member will be the context creator
OracleUserSecurity Admins	Special privileges for user entries. This group has read and write privileges for wallet password hints and passwords. The initial member is the person who creates the Oracle Context
OraclePasswordAccessibleDomains	The enterprise domains trusted to read the database password verifier of users, so that users can log in as password-authenticated global users. The initial (dummy) member is <code>OracleDBSecurityAdmins</code>
OracleDBCreators	Privileges to add new database entries under an Oracle Context. The first member of this group is the context creator
OracleDBAdmins	Full privileges for a given database and its subtree

Directory Deployment Factors for Oracle Advanced Security

When deploying Oracle Advanced Security features with Oracle Internet Directory, be sure to do the following:

- Centralize authentication and authorizations, so that an administrator need only delete a user in one place.

This feature revokes all of the user's privileges and minimizes the risk of retaining unintended privileges.

- Centralize security information, so that you can centralize security expertise.

Directory administrators knowledgeable about security manage directory security and user roles and privileges. This relieves DBAs of the burden of performing these functions. The net result is better security.

- Plan membership in enterprise domains carefully.

Be aware that current user database links operate only between databases within a single enterprise domain. Exercise care when you assign databases to a domain, because password authentication for enterprise users is defined at the domain level. Enterprise roles, too, are defined at the domain level. If you want databases to share an enterprise role, make sure that they are members of the same domain.

See Also: Chapter 15, "Managing Enterprise User Security," in *Oracle Advanced Security Administrator's Guide*

Application Context

Application Context is a database security feature that enables you to develop applications that are based on a user's session information. It provides a way to define, set, and access attributes that an application can use to enforce access control. Of the four types of Application Context—global, local, external, and centralized—the last, the context that is created using the "initialized globally" clause, uses Oracle Internet Directory

This section covers the following topics:

- [How Application Context Uses Oracle Internet Directory](#)
- [Application Context Entries Under the Oracle Context](#)
- [Security Measures for Application Context Entries](#)

How Application Context Uses Oracle Internet Directory

The user of an application context can have the attributes for an initial context, in the form of entries, set up for her in Oracle Internet Directory. If she successfully authenticates using Oracle Advanced Security, her global roles are retrieved from the directory; then her global application context is retrieved. By the time she logs on to the database, her global roles and initial application context are set up.

To understand how Application Context uses Oracle Internet Directory, consider the steps involved in setting up the hypothetical application context `HR`. Suppose that the application administrator would like to use this context to allow the user access to an application module called `HR`, which includes a personnel table. This user's information is stored in the directory, not in the personnel table. Nevertheless, the administrator will allow her restrictive access to the personnel table, using a PL/SQL procedure called `GetPersonnelData` to call the `HR` context.

1. The administrator creates a global user called `user1` in the database, using a DN to identify her as an enterprise user in the directory.
2. The administrator creates an application context in the database for the application `HR`, using a SQL command to implement a context package created using PL/SQL.
3. The administrator creates the directory entry `HR`, using an LDIF script. He assigns the subentries `Title` and `Manager` to the entry `HR`. He stores all of these entries within the domain `MyDomain`, which is located in the container `OracleDBAppContext`.

4. The administrator assigns the global user name `user1` as an attribute to the entry `Manager`.
5. The administrator writes a PL/SQL procedure—in this case, `GetPersonnelData`—that uses the application context to retrieve only those records with values matching the context.

When `user1` connects to a database belonging to the domain `myDomain`, her title is set to `Manager`, and any other information relating to her is retrieved from the LDAP directory. For instance, if her user entry contains the object class `inetOrgPerson`, attributes for this object class are retrieved.

When she executes the command `GetPersonnelData`, the user retrieves records only for persons whose title is `Manager`.

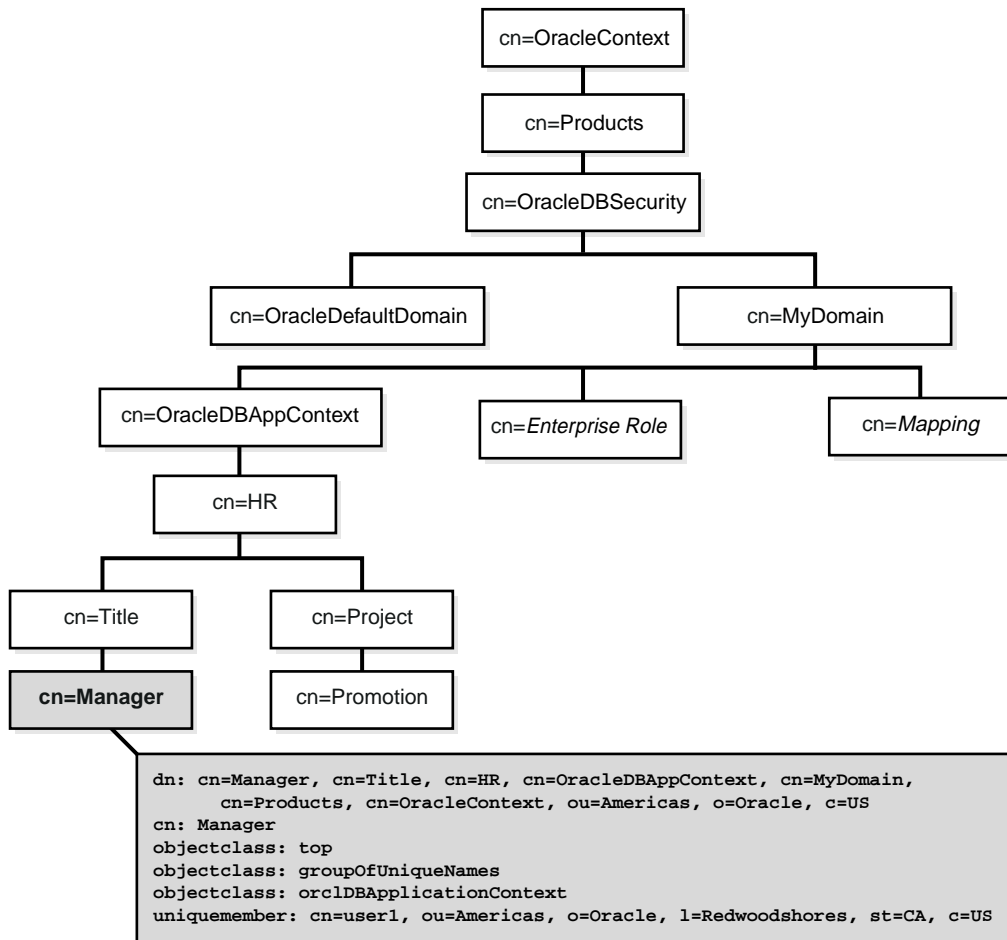
Application Context Entries Under the Oracle Context

As [Figure 4-6](#) on page 4-18 illustrates, a centrally initialized application context stores four types of entries in the directory:

- Context container—`OracleDBAppContext`
- Context namespace—in this case `HR`
- Context attribute—in this case `Title`
- Context value—in this case `Manager`

The values of the application context belong to the object class `orclDBApplicationContext`, which is a subclass of `groupOfUniqueNames`. Note that entries for Application Context are located within the container `OracleDBSecurity` under the enterprise domain to which the application context applies—in this case, `MyDomain`.

Figure 4–6 Directory Information Tree for Application Context, Showing Attributes for the Context Value



Security Measures for Application Context Entries

Directory entries for a centrally initialized Application Context are protected by Access Control Lists (ACLs) at two levels: at the level of the container `OracleDBSecurity` and at the level of the enterprise domain. At the first level, `OracleDBSecurityAdmins` have complete access to all enterprise domains and their subtrees. At the second level, `OracleDomainAdmins` have full access to application context values for their domain. For a context to work, all databases belonging to a domain must be able to read values belonging to contexts in that domain.

See Also: "Application Context Initialized Globally," in Chapter 12 of *Oracle9i Application Developer's Guide - Fundamentals*

Oracle Advanced Queuing

Oracle Advanced Queuing is a feature that combines a message queuing system with the Oracle database, using queue tables to store information about messages. This model facilitates persistent storage and message propagation between queues on different machines and databases.

Oracle Advanced Queuing uses different programmatic environments to provide two modes of message dissemination: point-to-point and publish-subscribe. In the first mode, senders and receivers use a common queue to exchange messages that have only one recipient. In the second, a message might be received by multiple recipients, called subscribers, who may subscribe to multiple queues located on different databases. These multi-consumer queues are called global topics.

This section covers the following topics:

- [How Oracle Advanced Queuing Uses Oracle Internet Directory](#)
- [Oracle Advanced Queuing Entries Under the Oracle Context](#)
- [Security Measures for Oracle Advanced Queuing Entries](#)
- [Directory Deployment Factors for Oracle Advanced Queuing](#)

How Oracle Advanced Queuing Uses Oracle Internet Directory

Oracle Advanced Queuing uses Oracle Internet Directory as a repository for the metadata of global topics and as a registry for database event notifications. In the first instance, connection factories and destinations for Java Messaging Service can be stored in a namespace accessible to Java Native Directory Interface (JNDI). In the second instance, clients can perform "open registration"—that is, they can use a single directory entry to register for multiple databases.

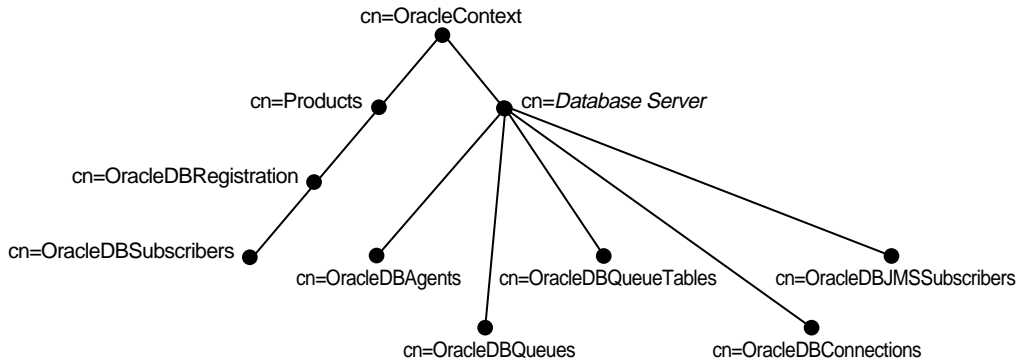
When a queue, queue table, or subscriber is created in a database, the database automatically creates directory entries that contain object metadata. For example, directory entries for queues contain information that references particular queue tables and indicates whether the corresponding queues are multiple consumer queues.

Using PL/SQL or Java interfaces, you can also add directory entries for aliases and JMS connection factories. The latter consist of the configuration parameters needed to establish a connection with a database.

Oracle Advanced Queuing Entries Under the Oracle Context

As [Figure 4-7](#) illustrates, Oracle Advanced Queuing stores entries for global topics directly below the database server entry to which they apply.

Figure 4-7 Directory Information Tree for Oracle Advanced Queuing



It uses five containers for this purpose, one for each of the object types needed to support global subscriptions. Aliases, too, are stored directly below database server entries. [Table 4-4](#) describes the contents of these five containers.

Table 4-4 Containers for Global Topics Entries

Container	Contents
cn=OracleDBAgents	Database agents
cn=OracleDBQueues	Queues. Subscribers of queues are placed under corresponding queue entries
cn=OracleDBQueueTables	Queue tables
cn=OracleDBConnections	Connection factories
cn=OracleDBJMSSubscribers	JMS subscribers. Contains detailed information about queue subscribers as well as links to subscriber entries

Client registrations for database event notifications have a container of their own, `cn=OracleDBRegistration`, which is located directly beneath the products container. Below `cn=OracleDBRegistration` is the entry `cn=OracleDBSubscribers`. This entry defines the LDAP users who are authorized to add, modify, and delete registration entries.

Security Measures for Oracle Advanced Queuing Entries

Everyone can read entries related to global topics, but only the database server that created these entries can modify them. For LDAP registration of event notifications, users who are granted the global role `global_aq_user_role` can add, modify, and delete registration entries.

Because global roles are implemented as privilege groups in Oracle9i, everyone granted an enterprise role that includes the global role `global_aq_user_role` is included in the privilege group `cn=OracleDBSubscribers`. Each database server is also a member of `cn=OracleDBSubscribers`.

Note that a registration entry can contain an ACI to ensure that only the entry creator and the database server can alter it.

Directory Deployment Factors for Oracle Advanced Queuing

Be sure to do the following before using Oracle Internet Directory with Oracle Advanced Queuing:

- Verify that each enterprise domain contains the enterprise role `enterprise_aq_user_role`.
- Verify that the privilege group `cn=orclDBSubscribers` is appropriately populated by checking that the database and `enterprise_aq_user_role` are attributes of this entry.
- Verify that the global role `global_aq_user_role` is an attribute of `enterprise_aq_user_role`.
- Verify that `enterprise_aq_user_role` is an attribute of `cn=OracleDBAQUsers`.
- If you move a database across enterprise domains, remove `global_aq_user_role` from `enterprise_aq_user_role` in the old domain. Add it to `enterprise_aq_user_role` in the new domain.

See Also: *Oracle9i Application Developer's Guide - Advanced Queuing*

Oracle Dynamic Services

Oracle Dynamic Services offers a programmatic framework for e-businesses to register and reuse existing Internet, Intranet, and database information services. It enables e-businesses to transform these services and tailor them to meet their own requirements.

Note: Oracle Dynamic Services is being deprecated beginning with Oracle9i Database Release 2 (9.2)

Starting in Oracle9iAS release 2 (9.0.2), Oracle is delivering an integrated, J2EE-compliant Web Services platform. Oracle Dynamic Services has been integrated with Oracle9iAS Web Services as the XML/HTML Stream Processing Tool. This tool allows Web service developers to present an HTML/XML source (such as a static Web page or HTML form) as an EJB and deploy to Oracle9iAS for use by J2EE client application developers. Furthermore, developers can expose their J2EE applications as SOAP-compliant Web services and register the corresponding Web service description (WSDL) in the supplied UDDI repository for discovery by external clients.

These features are described in the *Oracle9i Application Server Web Services Developer's Guide*, Part No. A95453-01. Oracle9iAS Release 2 provides a standards-based, fully integrated J2EE and Web services deployment platform. The current Dynamic Services functionality has been integrated into the Oracle9iAS platform, and the Dynamic Services terminal release will be delivered with Oracle9i Database Release 2 (9.2).

The Oracle Dynamic Services framework enables creation and aggregation of services from a variety of content sources on the Internet. Oracle Dynamic Services supports content access from:

- Databases using SQL and PL/SQL
- Remote service repositories using Simple Object Access Protocol (SOAP)
- Internet applications repositories using HTTP/HTTPS
- Miscellaneous application repositories using supported protocols extended or enhanced by application developers or any other extensible adapters.

The Oracle Dynamic Services framework supports service deployment anywhere over any protocol, including the following:

- Simple Mail Transfer Protocol (SMTP)

Inside the Oracle Dynamic Services framework, SMTP can be used to generate system or business messages.

- Wireless Application Protocol (WAP)

Results of service execution can be delivered to any mobile device.

E-businesses can use Oracle Dynamic Services in their database applications, hosted applications, online exchanges, and portals (B2B, B2C, B2M).

This section covers the following topics:

- [How Oracle Dynamic Services Uses Oracle Internet Directory](#)
- [Oracle Dynamic Services Entries Under the Oracle Context](#)
- [Security Measures for Oracle Dynamic Services Entries](#)
- [Directory Deployment Factors for Oracle Dynamic Services](#)

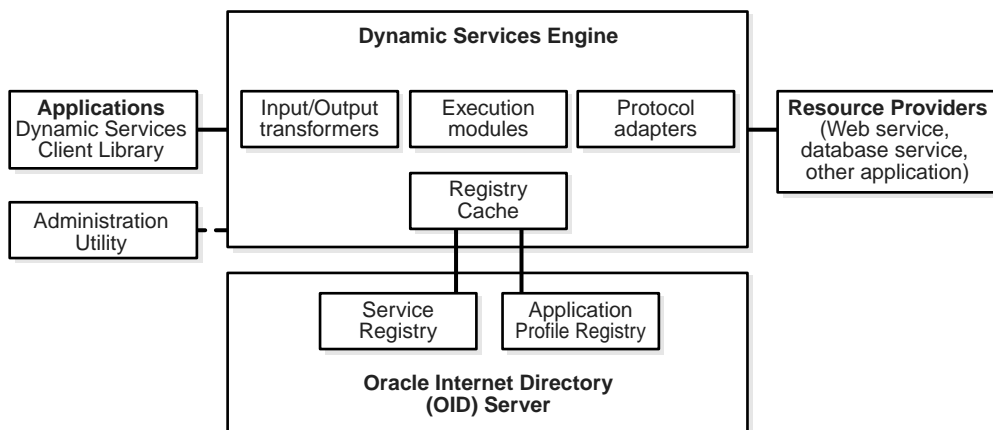
How Oracle Dynamic Services Uses Oracle Internet Directory

The Oracle Dynamic Services framework contains two registries, both directory based:

- Oracle Dynamic Services service registry (SR)
 - The service registry is the placeholder for all service definitions. Consumers can use the client library to query and update the service registry.
- Oracle Dynamic Services application profile registry (UPR)
 - The application profile registry is the placeholder for all validated Dynamic Services engine (DSE) applications, which are considered to be DS consumers. The registry stores access policies and application properties.

Figure 4–8 shows how these two registries interact with other components within the Oracle Dynamic Services framework.

Figure 4–8 LDAP Server Within Oracle Dynamic Services Framework Architecture



The Oracle Dynamic Services framework uses Oracle Internet Directory to store and manage service definitions and consumer profiles.

To avoid bottlenecking the directory and to increase performance, a DSE instance handles operations on the local registry cache first. The DSE instance notifies the directory server only if these operations modify the registry. Such modification might, for example, involve removing a service entry.

To ensure consistency between the registry cache and the central registries in the directory, the DSE instance updates the cache only after the directory performs the same action. This feature also ensures consistency between DSE instances.

Figure 4–9 on page 4-27 shows the synchronization process that occurs when an administrator registers the YahooQuote service, a new service for Oracle Dynamic Services, through one DSE instance.

1. The administrator connects to one DSE instance and registers the YahooQuote service, which has the unique service ID "urn:com.yahoo:quote" and which falls into the service category "business, finance, stock."
2. The DSE instance processes the service registration request, pre-registering the service package in its local service registry. If the pre-registration process is error free, the DSE instance sends the YahooQuote service package to the directory server for registration.
3. Oracle Internet Directory registers the YahooQuote package.
4. After the directory registers the YahooQuote service, it notifies the DSE instance. The DSE instance updates the local registry cache and informs the administrator that registration has been completed.

Figure 4–9 YahooQuote Service Registration

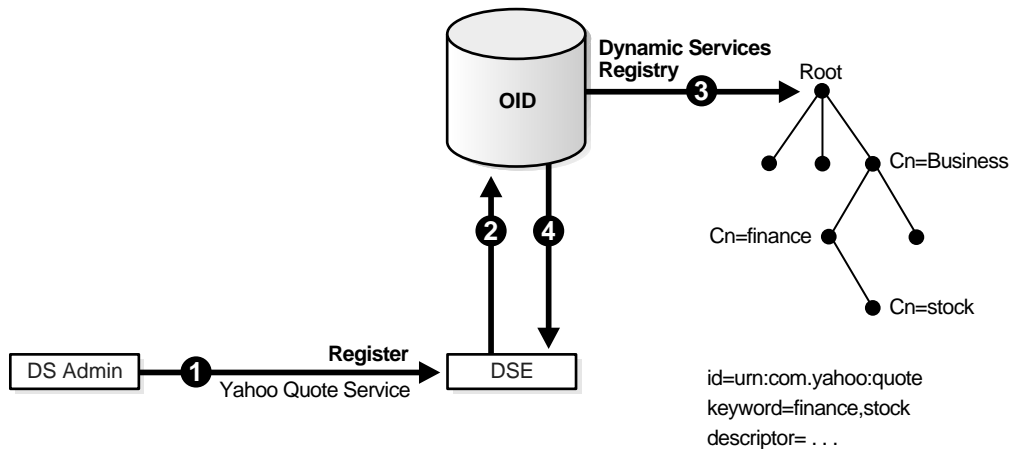
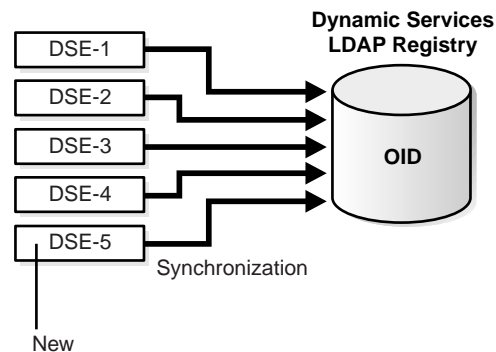


Figure 4–10 shows the synchronization process that occurs when the administrator starts a new DSE instance. During bootstrapping, this instance connects with the directory and synchronizes with the central registries.

Figure 4–10 Registry Synchronization Process for a New Dynamic Services Engine Instance



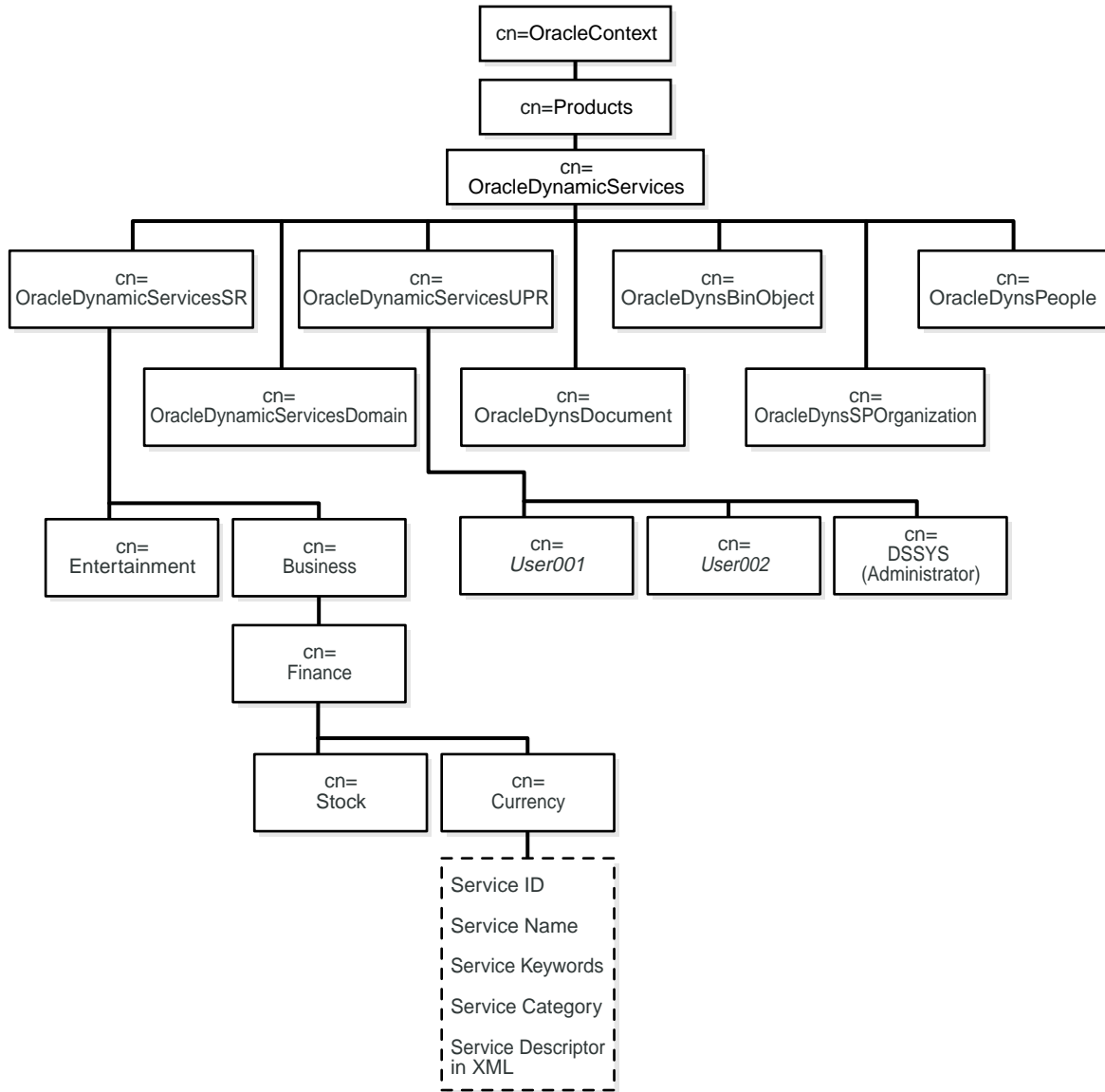
Oracle Dynamic Services Entries Under the Oracle Context

Oracle Dynamic Services stores the following entries in Oracle Internet Directory:

- `cn=OracleDynamicServicesSR`
The service registry for Oracle Dynamic Services. All service categories and registered services created within the Oracle Dynamic Services framework are stored under this entry
- `cn=OracleDynamicServicesUPR`
The application profile registry for Oracle Dynamic Services. Profiles for each valid Dynamic Services application are stored under this entry. These profiles contain information about service-specific properties and service access privileges
- `cn=OracleDynamicServicesDomain`
The entry that defines the scope of a set of DSE instances. Each DSE is represented under this entry. The entry contains a full set of attributes that describe properties such as connection and security for each engine instance
- `cn=OracleDynsDocument`
The subtree for service-related documents, such as service input schema and output schema. During the service registration process, a unique document ID is assigned to the document. At runtime, a document is retrieved by its document ID
- `cn=OracleDynsBinObject`
The entry under which all service-related binary files are stored. One of these files is the jar file included in the service package
- `cn=OracleDynsSPOrganization`
The subtree used to store organization profiles for service providers. Each profile includes entries for company name, company logo URL, and company Web site URL
- `cn=OracleDynsPeople`
The entry under which all contact information, such as e-mail addresses and phone numbers, is stored

Figure 4-11 on page 4-29 shows the structure of the directory subtree for Oracle Dynamic Services.

Figure 4–11 Directory Information Tree for Oracle Dynamic Services, Showing Attribute Types for One Service, Currency



Security Measures for Oracle Dynamic Services Entries

Oracle Dynamic Services grants full access (read/write) to the DSAdmin group, the users who have administrative privileges. Anonymous directory users have no access to the service and application profile registries.

Directory Deployment Factors for Oracle Dynamic Services

Consider the following factors before using Oracle Internet Directory with Oracle Dynamic Services:

- Registry entries for Oracle Dynamic Services are not designed for general use. They are dedicated exclusively to Oracle Dynamic Services, and only the DSAdmin group can access them, using a standard Java LDAP interface.
- Load balancing on the directory server is very important for Oracle Dynamic Services. Before setting up the directory server instance, estimate the level of server traffic carefully. As visits to the directory server increase, a replicated directory server instance may be required.
- Migration of registry data for Oracle Dynamic Services requires extra attention. It should be factored into directory design considerations.
- If no root Oracle Context is present in the directory when Oracle Dynamic Services is installed, the entry `cn=US` is designated as the default root.

Oracle Dynamic Services is certified to use Oracle Internet Directory—that is, its registry structure is compatible with this directory service. Oracle's LDAP Schema Council carefully reviews object classes and attributes for the product.

See Also: *Oracle Dynamic Services User's and Administrator's Guide*

Completing Directory Usage Configuration

This chapter describes the configuration steps that enable your Oracle home to use an Oracle Internet Directory server. The chapter describes the configuration steps common to all Oracle products; then it directs you to resources that describe directory configuration tasks particular to each Oracle product.

The chapter covers the following topics:

- [Prerequisites for Directory Use](#)
- [Options for Directory Usage Configuration](#)
- [Configuring Directory Usage After Installing the Database](#)
- [Configuring Directory Usage During a Custom Database Installation](#)
- [Configuring Directory Usage During a Client Installation](#)
- [Administrative Groups](#)
- [Product-Specific Configuration Tasks](#)

Prerequisites for Directory Use

Conceptually, there are five major prerequisites for an Oracle RDBMS to communicate with the directory:

- The directory must contain the current version of the Oracle Schema. The Oracle Schema contains all Oracle-specific object classes and attributes. It is preinstalled in Oracle Internet Directory.
- The directory must contain the current version of the Oracle Context. This is the subtree where all the Oracle-specific objects are stored. Oracle Internet Directory is shipped with a preinstalled Oracle Context at the directory root. You can create other Oracle Contexts elsewhere, but do not remove the root context.
- Your Oracle home must be configured for directory usage. Configuring your Oracle home entails setting parameters in the `ldap.ora` file that enable Oracle programs to connect to the directory. These parameters consist of values for directory host, port, and type and for the location of the Oracle Context.
- The directory must contain an entry for the database, so that the database can bind (log in) to the directory.
- Some directory-enabled features such as Oracle Advanced Security require the directory to be enabled for two-way authentication using Secure Sockets Layer (SSL). This feature requires that the user have an Oracle wallet in the directory.

The first two prerequisites are met by default if you are using the latest version of Oracle Internet Directory. If you are not using the latest version, Oracle Net Configuration Assistant updates both the Oracle Schema and the Oracle Context. This tool also creates the `ldap.ora` file. Database Configuration Assistant satisfies the fourth requirement: it creates an entry for the database in the directory, a process called database registration. Both tools must be run to complete directory usage configuration. The sections that follow explain how to run them.

To learn how to create and upload and download wallets from the directory, see Chapter 17, "Using Oracle Wallet Manager" in *Oracle Advanced Security Administrator's Guide*. To learn how to create and start an SSL instance when the directory is Oracle Internet Directory, see "Task 2: Start a Server Instance" in Chapter 3 of *Oracle Internet Directory Administrator's Guide*.

Options for Directory Usage Configuration

There are three methods for completing directory usage configuration:

1. Configure directory usage after installing a database. This method enables your Oracle home to use a directory
2. Configure directory usage when you install and register a database. This method is part of a custom database installation. It is an alternative to the first method.
3. Configure your client to use a directory to connect to a database. This method is part of a client installation

Configuring Directory Usage After Installing the Database

Oracle Net Configuration Assistant and Database Configuration Assistant can be used to complete directory usage configuration at any time. If you choose this option, both tools must be run in standalone mode. The first tool enables you to choose a directory server. The second registers your database—that is, it creates an entry for the database in the directory. Without this entry, your Oracle home cannot access the directory.

This section covers the following topics:

- [Using Oracle Net Configuration Assistant to Configure Directory Usage](#)
- [Using Database Configuration Assistant to Register a Database](#)

Using Oracle Net Configuration Assistant to Configure Directory Usage

To configure directory server usage:

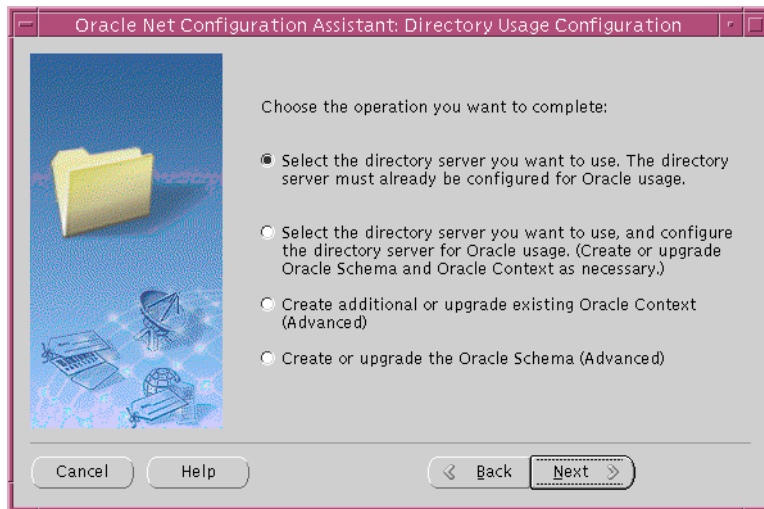
1. Start Oracle Net Configuration Assistant:
 - On UNIX, go to `$ORACLE_HOME/bin`; then enter the command `netca`
 - On Windows NT, choose **Start > Programs > Oracle - HOME_NAME > Configuration and Migration Tools > Net Configuration Assistant**

The Welcome page appears.

2. Select Directory Service Usage Configuration, and then choose **Next**.

The Directory Usage Configuration page is shown in [Figure 5-1](#) on page 5-4.

Figure 5–1 Oracle Net Configuration Assistant: Directory Usage Configuration Page



3. Select one of the four options on this page; then follow the prompts in the wizard and Help to complete directory usage configuration.

The options are as follows:

Option 1: Select the directory server you want to use

Select this option to enable your Oracle home to use a directory server that is already configured to use directory-enabled features.

Once configuration is complete, this option enables your computer to look up entries in the directory. It prompts you to do the following:

- Select the type of directory server.
- Identify the host name and port of the directory server. The default non-SSL port number is 389. The default SSL port is 636.
- Select a directory entry that contains an Oracle Context from which this server can access Oracle entries.

Option 2: Select the directory server you want to use, and configure the directory server for Oracle usage

Select this option to configure a directory server for directory-enabled features and to enable your Oracle home to use that directory. This option is designed for administrators who are configuring these features for the first time.

Once configuration is complete, this Oracle home can look up entries in the directory.

The options are the same as those for Option 1. The difference is that, if the Oracle Schema does not exist or is an older version, you are prompted to create it or upgrade it. Having the correct schema version is a prerequisite for creating or designating an Oracle Context. There are three possible options for choosing an Oracle Context:

1. Accept the root Oracle Context as your default. The root Oracle Context is at the root entry, or top entry, of a directory.
2. Choose from a drop-down list of Oracle Contexts.
3. If no Oracle Context exists, create one under a directory entry of your choice. To add an Oracle Context, you must use Option 3. Note that some Oracle features require a root Oracle Context—that is, one that is located at the root entry of a directory. If the root Oracle Context is missing, you can create one by selecting "root entry" from the drop-down list of directory entries that Oracle Net Configuration Assistant displays.

Option 3: Create additional or upgrade existing Oracle Context

You must select this option to add an Oracle Context to your directory if it already contains an Oracle Context. In addition, this option can be used to upgrade an old version of the context.

To create an Oracle Context, the following must exist in the directory server:

- A directory entry under which you want the Oracle Context to be created.
- The current Oracle Schema.

If the Oracle Context is an older version, you are prompted to upgrade it. This is important because an Oracle9i database will not work with an Oracle8i Oracle Context or an earlier one. You can use the upgraded Oracle Context to register any Oracle8i databases that are created in the future.

Option 4: Create or upgrade the Oracle Schema

In the unusual event that Oracle Internet Directory contains no Oracle Schema, you can use Option 4 to create the Schema without having to resort to the full directory usage configuration explained in Options 1 and 2. In addition, Option 4 can be used to upgrade the Oracle Schema.

Notes:

- To create an Oracle Context, you must have credentials that enable you to create a new subtree under the entry chosen for the new context. This means either that you are a directory administrator or that an ACL somewhere in the directory enables you to modify the entry.
 - To update the Oracle Schema, you must have the credentials of a directory administrator.
 - If, later, you choose a different Oracle Context or create a new one, remember to reregister your database under this context.
 - You cannot complete directory usage configuration as part of an Enterprise Edition or Standard Edition installation on the server. If you choose these installation options, you must run Oracle Net Configuration Assistant in standalone mode.
-
-

Using Database Configuration Assistant to Register a Database

After running the Oracle Net Configuration Assistant, run the Database Configuration Assistant to register your database in the directory. To register a database, you must be a member of either the Database Registration group or the OracleContextAdmins group, or you must be the directory superuser. Use Oracle Enterprise Security Manager to add administrators to these two groups. To learn how to use this tool see Chapter 18, "Using Oracle Enterprise Security Manager", in *Oracle Advanced Security Administrator's Guide*. Note that, if you are using Enterprise User Security, you can use Enterprise Security Manager to register a database.

To register a database in the directory, using Database Configuration Assistant in standalone mode:

1. Start the Database Configuration Assistant as follows:
 - Windows NT: Select **Start->Programs->Oracle-<Oracle-HOME_NAME>->Database Administration->Database Configuration Assistant**
 - UNIX: Select `$ORACLE_HOME/bin/dbca`
2. Select **Configure database options in a database** and choose **Next**.
3. Select a database and choose **Next**.

The final Database Configuration Assistant window appears.

4. Choose **Yes, Register the Database**, and enter the directory credentials for a user in the Database Registration group.
5. Choose **Finish** if you are just registering a database; choose **Next** if you want to select additional database features.
6. If you chose **Finish**; the Locate Initialization File window appears.
7. Select the appropriate initialization file and choose **OK**.

If you execute these steps correctly, Database Configuration Assistant does the following:

- Creates a new database service object and subtree underneath your chosen Oracle Context in the directory
- Adds the database to the default enterprise domain. (To learn about enterprise domains, see "[Oracle Advanced Security Entries Under the Oracle Context](#)" in Chapter 4)

- Adds the distinguished name of the database to the database server parameter file, `spfile.ora`, as an `RDBMS_SERVER_DN` initialization parameter value
- Restarts the database, so that the new initialization parameter takes effect

Configuring Directory Usage During a Custom Database Installation

After installing database server software, Oracle Universal Installer launches Oracle Net Configuration Assistant, which gives you the option of completing directory usage configuration. Completing configuration consists of the following:

- Selecting a directory type.
- Specifying the directory's host name and port. The default port numbers are 389 for non-SSL connections and 636 for SSL connections.
- Selecting a directory entry that contains an Oracle Context.

If the required Oracle Schema is already installed, Oracle Net Configuration Assistant prompts you to select an Oracle Context from a drop-down list of directory entries. If it was created during directory setup, one of the entries in the list is a root Oracle Context. The root Oracle Context is at the root entry, or top entry, of a directory.

If only the root context is present, you can either use this context or create a new Oracle Context by running Oracle Net Configuration Assistant in standalone mode. (See "[Option 3: Create additional or upgrade existing Oracle Context](#)".)

If no root Oracle Context is present, you can create one by selecting "root entry" from the drop-down list of directory entries. Note that some Oracle features require that a root Oracle Context be present.

If the required Oracle Schema is not installed, Oracle Net Configuration Assistant gives you the option of installing the correct schema or deferring directory configuration until a later time.

If you perform a custom database installation, Database Configuration Assistant runs automatically, after Oracle Net Configuration Assistant. It prompts you to register the database. Choose **Yes, Register the Database**, and then follow steps 4 through 7 in "[Using Database Configuration Assistant to Register a Database](#)".

Administrative Groups

The administrator who successfully creates or updates an Oracle Context in the course of completing directory configuration is automatically added to five administrative groups:

- **OracleContextAdmins**
(cn=OracleContextAdmins , cn=Groups , cn=OracleContext)
This group has full privileges for the entire Oracle Context.
- **OracleDBCreators** (cn=OracleDBCreators , cn=OracleContext)
This group enables you to use Database Configuration Assistant to register a database service entry in the directory together with its connect descriptor.
- **OracleNetAdmins** (cn=OracleNetAdmins , cn=OracleContext)
This group can use Oracle Net Manager to create, modify, and delete net service names and to modify Oracle Net attributes of database services.
- **OracleDBSecurityAdmins**
(cn=OracleDBSecurityAdmins , cn=OracleContext)
This group has full privileges over directory objects in the container `OracleDBSecurity`. These objects consist of enterprise domains, enterprise roles, and mappings between users and shared database schemas.
- **OracleUserSecurityAdmins**
(cn=OracleUserSecurityAdmins , cn=Groups , cn=OracleContext)
This group has read and write privileges for wallet password hints and passwords.

Configuring Directory Usage During a Client Installation

A client installation gives you the option of using database services, net service names, or net service aliases stored in the directory to connect to a database. This feature is called directory naming. If you choose to use the directory to connect to a database, Oracle Net Configuration Assistant prompts you to do the following:

- Select a directory type.
- Specify the directory's host name or port. The default port numbers are 389 for non-SSL connections and 636 for SSL connections.
- Select a directory entry that contains an Oracle Context.

If the Oracle Schema is incorrect or was not installed or no Oracle Context is present, you cannot complete directory usage configuration on the client. To complete configuration, run Oracle Net Configuration Assistant in standalone mode after installing the database.

For more about database services, net service names, and net service aliases, see "[Oracle Net Services Entries Under the Oracle Context](#)" in Chapter 4, "Deploying Oracle Products with Oracle Internet Directory."

Product-Specific Configuration Tasks

Oracle Net Configuration Assistant performs only the minimal directory configuration tasks necessary for most Oracle products. As such, many directory-enabled Oracle products may require additional configuration. [Table 5-1](#) lists each product described in this book and provides links to documents that describe product-specific configuration tasks.

Table 5-1 Links to Product-Specific Configuration Information

Product	Document
Oracle Net Services	<i>Oracle9i Net Services Administrator's Guide</i> , Chapter 8, "Setting Up Directory Server Usage"
Oracle Advanced Security	<i>Oracle Advanced Security Administrator's Guide</i> , Chapter 15, "Managing Enterprise User Security"
Application Context	<i>Oracle9i Application Developer's Guide - Fundamentals</i> , "Application Context Initialized Globally", in Chapter 12, "Policy-Based Security"
Oracle Advanced Queuing	<i>Oracle9i Application Developer's Guide - Advanced Queuing</i> , Chapter 12, "Creating Applications Using JMS"
Oracle Dynamic Services	<i>Oracle Dynamic Services User's and Administrator's Guide</i> , "Using Lightweight Directory Access Protocol (LDAP) as a Master Registry," in Chapter 4, "Advanced Installation Options"

Oracle-Specific LDAP Schema Extensions

This appendix provides a comprehensive list of the structural object classes and attributes that LDAP-enabled Oracle products use to define entries in Oracle Internet Directory. Except in the case of Oracle9iAS Email, which uses auxiliary object classes, each attribute listed is cross-referenced with its object class or, if it belongs to more than one object class, with its object classes. The appendix groups object classes and attributes by product.

The appendix covers the following products:

- [Oracle Net Services](#)
- [Oracle Advanced Security](#)
- [Application Context](#)
- [Oracle Advanced Queuing](#)
- [Oracle Dynamic Services](#)

Oracle Net Services

This section lists and describes structural object classes and attributes for Oracle Net Services.

Structural Object Classes

- `orclDBServer`
Defines the attributes for database service entries
- `orclNetService`
Defines the attributes for net service name entries
- `orclNetServiceAlias`
Defines the attributes for net service alias entries
- `orclNetDescription`
Specifies a connect descriptor, which contains the listener protocol address and the connect information to the service
- `orclNetDescriptionList`
Specifies a list of connect descriptors
- `orclNetAddress`
Specifies a listener protocol address
- `orclNetAddressList`
Specifies a list of addresses

Attributes

- `orclNetAddrList` (`orclNetAddressList`, `orclNetDescription`)
Identifies one or more listener protocol addresses
- `orclNetAddressString` (`orclNetAddress`)
Defines a listener protocol address
- `orclNetConnParamList` (`orclNetDescription`)
Placeholder for future connect data parameters

- `orclNetDescList` (`orclNetDescriptionList`)
Identifies one or more connect descriptors
- `orclNetDescName` (`orclDBServer`, `orclNetService`)
Identifies a connect descriptor or a list of connect descriptors
- `orclNetFailover` (`orclNetDescription`, `orclNetAddressList`)
Turns connect-time failover on for an address list
- `orclNetInstanceName` (`orclNetDescription`)
Specifies the instance name to access
- `orclNetLoadBalance` (`orclNetDescription`, `orclNetAddressList`)
Turns client load balancing on for an address list
- `orclNetProtocol` (`orclNetAddress`)
Identifies the protocol used in the `orclAddressString` attribute
- `orclNetSdu` (`orclNetDescription`)
Specifies the session data unit (SDU) size
- `orclNetServiceName` (`orclNetDescription`)
Specifies the Oracle9i or Oracle8i database service name in the `CONNECT_DATA` portion
- `orclNetSourceRoute` (`orclNetDescription`, `orclNetAddressList`)
Instructs Oracle Net to use each address in order until the destination is reached
- `orclSid` (`orclNetDescription`)
Specifies the Oracle System Identifier (SID) in the `CONNECT_DATA` portion of a connection descriptor
- `orclVersion` (all object classes)
Specifies the version of software used to create the entry

Oracle Advanced Security

This section lists and describes structural object classes and attributes for Oracle Advanced Security.

Structural Object Classes

- `orclDBEnterpriseDomain`
A group object class that identifies the database members of a domain. Contains the domain's configuration data. For example, it specifies the authentication types allowed and indicates whether current user database links are enabled
- `orclDBEnterpriseRole`
A group object class that defines an enterprise role within a domain as well as the users and database global roles assigned to this enterprise role
- `orclDBEntryLevelMapping`
Defines a single mapping between a user and a database schema
- `orclDBSubtreeLevelMapping`
Defines a mapping between a user subtree and a database schema

Attributes

- `uniquemember1` (`orclDBEnterpriseDomain`, `orclDBEnterpriseRole`)
Stores a list of databases that are members of a domain. Stores a list of users that are assigned to an enterprise role
- `orclDBAuthTypes` (`orclDBEnterpriseDomain`)
Indicates the type of user authentication that databases should accept
- `orclDBTrustedDomain` (`orclDBEnterpriseDomain`)
Indicates whether current user database links function between databases in the domain

¹ Oracle9i databases only. Oracle8i databases use `OracleDBServerMember` and `orclDBRoleOccupant`.

- `orclDBServerRole` (`orclDBEnterpriseRole`)
Defines a list of global roles for the databases within the domain
- `orclDBDistinguishedName` (`orclDBEntryLevelMapping`,
`orclDBSubtreeLevelMapping`)
Specifies the full distinguished name of the enterprise user
- `orclDBNativeUser` (`orclDBEntryLevelMapping`,
`orclDBSubtreeLevelMapping`)
Specifies the database shared schema name

Application Context

Application Context uses one structural object class, `orclDBApplicationContext`, to define context values. This object class uses one attribute, `uniquemember`, to define context users. `orclDBApplicationContext` is a subclass of `GroupOfUniqueNames`.

Oracle Advanced Queuing

This section lists and describes structural object classes and attributes for Oracle Advanced Queuing.

Structural Object Classes

- `orclDBAQConnection`
Stores an Advanced Queuing JMS connection factory object
- `orclDBAQObject`
Stores queues, queue tables, aliases, subscribers, JMS subscribers, and agents
- `orclDBAQRegistration`
Stores a registration request from an enterprise user

Attributes

- `orclDBAQGeneric` (`OrclDBAQConnection`)
Stores miscellaneous name value pairs
- `orclDBAQObjName` (`orclDBAQObject`)
Stores name of an Advanced Queuing object
- `orclDBAQObjOwner` (`orclDBAQObject`)
Stores name of the database user who owns the Advanced Queuing object
- `orclDBAQObjType` (`orclDBAQObject`)
Stores the type of Advanced Queuing object
- `orclDBAQPointerAttr` (`orclDBAQObject`)
 - The distinguished name of the underlying queue table for a queue object
 - The LDAP entry that contains the digital certificate of the agent
 - The distinguished name of the agent for a corresponding queue subscriber
 - The distinguished name of the queue subscriber for a corresponding JMS subscriber
 - The distinguished name of an aliased object

- `orclDBAQRegNamespace` (`orclDBAQRegistration`)
Stores the registration namespace, for example `AQ` or `anonymous`
- `orclDBAQRegSubscription` (`orclDBAQRegistration`)
Stores the subscription name of the registration
- `orclDBAQRegLocation` (`orclDBAQRegistration`)
Stores the location where the server sends notification
- `orclDBAQRegUser` (`orclDBAQRegistration`)
Stores the name of the enterprise user who is registering
- `orclDBAQRegUserContext` (`orclDBAQRegistration`)
Specifies the user context to pass back to the user when notification occurs
- `orclDBAQRegServers` (`orclDBAQRegistration`)
Specifies the distinguished names of the database servers in which the enterprise user is registering
- `orclDBAQRegUnreachable` (`orclDBAQRegistration`)
Specifies the distinguished names of the database servers that could not reach the client. Oracle database servers provide this value
- `orclDBAQRegRejected` (`orclDBAQRegistration`)
Specifies the distinguished names of the database servers that have rejected this registration request. Oracle database servers provide this value

Oracle Dynamic Services

This section lists and describes structural object classes and attributes for Oracle Dynamic Services.

Structural Object Classes

- `orclDyNsAccessibleService`
Specifies the services available to an application. Used in the subtree `OracleDynamicServicesUPR`
- `orclDyNsServiceModProperty`
Specifies application properties. Used in the subtree `OracleDynamicServicesUPR`
- `orclDyNsTxtObject`
Specifies placeholders for text documents under the `OracleDyNsDocument` subtree
- `orclDyNsBinObject`
Specifies placeholders for the binary files of a service. Used under `DSBBinObject` subtree
- `orclDyNsServiceCat`
Specifies a particular service category, such as "Business" or "Finance." Used in the subtree `OracleDynamicServicesSR`
- `orclDyNsServiceRegistryEntry`
Specifies an actual service entity. Used in `OracleDynamicServicesSR` subtree. Service ID, keywords, and interface are extracted to support service searches
- `orclDyNsEnterpriseDomain`
Specifies the DSE domain administrator, who manages all DSE instances within a domain. This object class is extensible for future implementation
- `orclDyNsEnterpriseInstance`
Specifies a DSE instance. Specifies how a connection is made to the DSE instance

- `orclDynsPerson`
Specifies the person who is the service contact
- `orclDynsSPOrganization`
Specifies the organization that is the service provider. Each trademarked organization name is a unique ID

Attributes

- `orclDynsBinaryHolder (orclDynsBinObject)`
Stores binary objects
- `orclDynsTextHolder (orclDynsTxtObject, orclDynsServiceRegistryEntry)`
Stores text documents
- `orclDynsObjRefCnt (orclDynsTxtObject, orclDynsBinObject, orclDynsSPOrganization)`
A counter used to track object references
- `orclDynsPropertyName (orclDynsServiceModProperty)`
Specifies the property name that an application uses for a service
- `orclDynsPropertyValue (orclDynsPropertyValue)`
Identifies a property value
- `orclDynsInternalObjectID (orclDynsTxtObject, orclDynsBinObject)`
Specifies a unique ID for references to internal objects, such as binary objects
- `orclDynsInternalObjectType (orclDynsTxtObject, orclDynsBinObject)`
Used for extensibility purposes
- `orclDynsKeywords (orclDynsServiceRegistryEntry)`
Specifies keywords for a service
- `orclDynsServiceID (orclDynsAccessibleService, orclDynsServiceRegistryEntry)`
Specifies a service ID

- `orclDynsServiceName` (`orclDynsServiceRegistryEntry`)
Specifies a service name
- `orclDynsModifier` (`orclDynsServiceModProperty`)
Used in `OracleDynamicServicesUPR` subtree as `<mod, pn, pv>`
- `orclDynsURL` (`orclDynsSPOrganization`)
Specifies a corporation's URL. One property of the service provider
- `orclDynsLogoURL` (`orclDynsSPOrganization`)
Specifies the logo of the corporation that is the service provider
- `orclDynsCopyright` (`orclDynsSPOrganization`)
Specifies the copyright for a service. One property of the service provider
- `orclDynsDomainAdminPassword` (`orclDynsEnterpriseDomain`)
Specifies the password for a domain administrator
- `orclDynsConnection` (`orclDynsEnterpriseInstance`)
Connection string for a DSE
- `orclDynsInterface` (`orclDynsServiceRegistryEntry`)
Specifies the interface for a particular service

LDAP Command-Line Tools

LDAP protocol operations are divided into three categories: authentication, interrogation, and update and control. The LDAP C-API provides a number of simple command-line tools that together cover all three categories.

The appendix covers the following topics:

- [LDAP Command-Line Tools](#)
- [Optional Arguments for Command-Line Tools](#)

LDAP Command-Line Tools

This section introduces six popular command-line tools. The section "[Optional Arguments for Command-Line Tools](#)", immediately following, defines the optional arguments used in the command descriptions and examples.

These are the six commands:

- [ldapbind](#)
- [ldapsearch](#)
- [ldapadd](#)
- [ldapdelete](#)
- [ldapmodify](#)
- [ldapmoddn](#)

ldapbind

Use the command-line tool `ldapbind` to authenticate to a directory server. You can also use `ldapbind` to find out if the server is running.

Syntax

```
ldapbind [options]
```

Example

```
ldapbind -h myhost -p 389 -D "cn=orcladmin" -w welcome
```

This command authenticates user `orcladmin` to the directory server `myhost` located at port `389`, using the password `welcome`.

ldapsearch

Use the command-line tool `ldapsearch` to search for specific entries in a directory. `ldapsearch` opens a connection to a directory, authenticates the user performing the operation, searches for the specified entry, and prints the result in a format that the user specifies.

Syntax

```
ldapsearch [options] filter [attributes]
```

Example

```
ldapsearch -h myhost -p 389 -s base -b "ou=people,dc=acme,dc=com" \  
"objectclass=*"
```

This command searches the directory server `myhost`, located at port 389. The scope of the search (`-s`) is `base`, and the part of the directory searched is the base DN (`-b`) designated. The search filter `"objectclass=*"` means that values for all of the entry's object classes are returned. No attributes are returned because they have not been requested. The example assumes anonymous authentication because authentication options are not specified.

ldapadd

Use the command-line tool `ldapadd` to add entries to the directory. `ldapadd` opens a connection to the directory and authenticates the user. Then it opens the LDIF file supplied as an argument and adds, in succession, each entry in the file.

Syntax

```
ldapadd [options] [-f LDIF-filename]
```

Example

```
ldapadd -h myhost -p 389 -D "cn=orcladmin" -w welcome -f jhay.ldif
```

Using this command, user `orcladmin` authenticates to the directory `myhost`, located at port 389. The command then opens the file `jhay.ldif` and adds its contents to the directory. The file might, for example, add the entry `uid=jhay,cn=Human Resources,cn=acme,dc=com` and its object classes and attributes.

See Also: ["LDIF"](#) on page 2-7 for details about LDIF file syntax

ldapdelete

Use the command-line tool `ldapdelete` to remove leaf entries from a directory. `ldapdelete` opens a connection to a directory server and authenticates the user. Then it deletes specified entries.

Syntax

```
ldapdelete [options] "entry DN"
```

Example

```
ldapdelete -h myhost -p 389 -D "cn=orcladmin" -w welcome \  
"uid=hricard,ou=sales,ou=people,dc=acme,dc=com"
```

This command authenticates user `orcladmin` to the directory `myhost`, using the password `welcome`. Then it deletes the entry `uid=hricard,ou=sales,ou=people,dc=acme,dc=com`.

ldapmodify

Use the command-line tool `ldapmodify` to modify existing entries. `ldapmodify` opens a connection to the directory and authenticates the user. Then it opens the LDIF file supplied as an argument and modifies the LDAP entries specified by the file.

`ldapmodify` uses a modified form of an LDIF file. Within the file itself, you use the attribute `changetype` to specify the type of change. An example is `changetype: add`.

Four types of changes are possible:

- `add`—adds a new entry
- `modify`—changes an existing entry, that is, it adds, deletes, or replaces attributes of the entry
- `delete`—deletes an existing entry
- `modrdn`—modifies the RDN of an existing entry

Syntax

```
ldapmodify [options] [-f LDIF-filename]
```

Example

```
ldapmodify -h myhost -p 389 -D "cn=orcladmin" -w welcome -f hricard.ldif
```

Using this command, user `orcladmin` authenticates to the directory `myhost`, located at port 389. The command then opens the file `hricard.ldif` and modifies the directory entries specified by the file. The file might, for example, change the telephone number attribute of entry

```
uid=hricard,cn=sales,cn=acme,dc=com.
```

Note: You can use `ldapmodify` instead of `ldapadd` and `ldapdelete` to add or delete entries.

ldapmoddn

Use the command-line tool `ldapmoddn` to:

- change the RDN of an entry
- move an entry or subtree to another location in the directory

Syntax

```
ldapmoddn [options] -b "current DN" -R "new RDN" -N "new Parent"
```

Example

```
ldapmoddn -h myhost -p 389 -D "cn=orcladmin" -w welcome \  
-b "uid=oball,ou=sales,ou=people,dc=acme,dc=com" \  
-N "ou=marketing,ou=people,dc=acme,dc=com"
```

This command authenticates user `orcladmin` to the directory `myhost`, using the password `welcome`. Then it assigns to the entry `uid=oball,ou=sales,ou=people,dc=acme,dc=com` a new parent entry, `ou=marketing,ou=people,dc=acme,dc=com`.

Optional Arguments for Command-Line Tools

[Table B-1](#) defines the optional arguments used in the command descriptions and examples.

Table B-1 Commonly Used Command-Line Options

Option	Description
-h	The host name of the directory server
-p	The port number of the directory server
-D	The bind DN—that is, the user authenticating to the directory
-w	The bind password in simple authentication
-W	Wallet location for one- or two-way SSL authentication
-P	Wallet password

Table B-1 Commonly Used Command-Line Options

Option	Description
-U	SSL authentication mode: <ul style="list-style-type: none"> ■ 1 for no authentication ■ 2 for one-way authentication ■ 3 for two-way authentication
-b ¹	The base DN for a search:
-s ²	Search scope: <ul style="list-style-type: none"> ■ base—the entry requested ■ one—the entries just below the requested entry ■ sub—the entire subtree
-f	The LDIF file containing additions, deletions, or modifications
-R	New RDN
-N	New parent for an entry or subtree that is moved

¹ Mandatory for `ldapsearch`

² Mandatory for `ldapsearch`

See Also: Chapter 5, "Command-Line Tools Syntax" in *Oracle Internet Directory Application Developer's Guide*

Index

A

- abstract object classes, 2-15
- access control items, 2-17
- access control lists. *See* ACLs
- ACIs, 2-17
- ACLs
 - Application Context, 4-19
 - examples, 2-18
 - Oracle Advanced Security, 4-14
 - placement, 3-7
 - structure, 2-17
- administrative groups
 - Groups container, 2-21
 - OracleDBAdmins group, 4-14
 - OracleDBC creators, 4-14
 - OracleDBSecurityAdmins, 4-14
 - OracleDomainAdmins, 4-14
 - OraclePasswordAccessibleDomains, 4-14
 - OracleUserSecurityAdmins, 4-14
- Application Context
 - ACLs, 4-19
 - attributes, A-5
 - directory entries, 4-17
 - directory information tree, 4-18
 - example, 4-16
 - object classes, A-5
 - overview, 1-2
 - product summary, 4-16
 - security measures, 4-19
- Application Context initialized centrally, 4-16
- application-specific directories
 - drawbacks, 2-4
 - features, 2-4

- attribute matching rules
 - definition, 2-12
 - example, 2-13
- attribute syntax
 - definition, 2-12
 - example, 2-13
- attributes
 - Application Context, A-5
 - definition, 2-11
 - examples, 2-11
 - foreign language, 2-13
 - operational, 2-11
 - Oracle Advanced Queuing, A-6, A-7
 - Oracle Dynamic Services, A-9, A-10
 - Oracle Net Services, A-2, A-3
 - user, 2-11
- authentication
 - anonymous, 2-17
 - simple, 2-17
 - simple over SSL, 2-17
 - SSL with certificates, 2-17
- auxiliary object classes, 2-14, 2-15

B

- backup and recovery, of directories, 3-4

C

- C LDAP API, 2-7
- command-line tools
 - ldapadd, B-5
 - ldapbind, B-3
 - ldapdelete, B-6

- ldapmoddn, B-8
- ldapmodify, B-7
- ldapsearch, B-4
- overview, 2-7

D

- Database Configuration Assistant, 2-19, 5-7
- database event notifications, 4-20
- database service registration, 5-7
- database services (database connect descriptors), 4-5
- databases
 - comparison with directories, 2-2 to 2-4
 - data units, 2-3
 - distribution, 2-3
 - entry format, 2-4
 - event notifications, 4-20
 - read-to-write ratio, 2-3
 - user authentication, 4-11
 - user authorization, 4-11
- directories
 - access control, 3-7
 - ACL placement, 3-7
 - ACLs, 2-17
 - applications, 2-4
 - application-specific, 2-4
 - attribute matching rules, 2-12, 2-13
 - attribute syntax, 2-12, 2-13
 - attributes, 2-11
 - authentication, 2-17
 - backup and recovery, 3-4
 - benefits, 2-4, 2-5
 - command-line tools, 2-7
 - comparison with databases, 2-2 to 2-4
 - configuration
 - after database installation, 5-3 to 5-4
 - client installation, 5-9
 - Oracle Context, 2-19
 - configuration tools
 - Database Configuration Assistant, 5-7
 - Oracle Net Configuration Assistant, 5-3 to 5-4
 - data units, 2-3
 - deployment factors, 4-10, 4-15, 4-22, 4-30
 - directory information trees, 2-9
 - distinguished names, 2-9
 - distribution, 2-3
 - entries, 2-9, 2-10, 2-11
 - entry format, 2-4
 - extensibility, 2-6
 - features, 2-9
 - Globalization Support, 2-6
 - information flow, 2-8
 - load estimation, 3-5
 - modification, 2-7
 - namespace design, 3-2, 3-3
 - partitions, 2-6, 3-4
 - read-to-write ratio, 2-3
 - referrals, 2-6
 - relative distinguished names, 2-9
 - replication, 3-3
 - schema, 2-16
 - schema discovery, 2-6
 - security, 2-6
 - standards, 2-4, 2-5
 - system requirements, 3-5
 - testing, 3-6
- directory applications, examples of, 2-4
- directory deployment factors
 - Oracle Advanced Queuing, 4-22
 - Oracle Advanced Security, 4-15
 - Oracle Dynamic Services, 4-30
 - Oracle Net Services, 4-10
- directory entries
 - Application Context, 4-17
 - attributes, 2-11
 - definition, 2-9, 2-10
 - examples, 2-2, 2-11
 - Oracle Advanced Queuing, 4-20, 4-21
 - Oracle Advanced Security, 4-12
 - Oracle Dynamic Services, 4-28
 - Oracle Net Services, 4-5
 - suitability, 3-2
- directory information trees
 - Application Context, 4-18
 - design, 3-2, 3-3
 - naming contexts, 2-15
 - Oracle Advanced Queuing, 4-21
 - Oracle Advanced Security, 4-13

- Oracle Dynamic Services, 4-29
- Oracle Net Services, 4-6
- directory interoperability, 1-2, 1-5
- directory migration
 - from Oracle Names servers, 4-9, 4-10
 - from tnsnames.ora file, 4-9, 4-10
 - to Oracle Names LDAP Proxy servers, 4-9, 4-10
- directory naming, 4-3
 - Java Database Connectivity (JDBC) Thin drivers, 4-4
 - OracleContextAdmins group, 4-9
 - OracleDBCreators group, 4-9
- directory security
 - Application Context, 4-19
 - Oracle Advanced Queuing, 4-22
 - Oracle Advanced Security, 4-14
 - Oracle Dynamic Services, 4-30
 - Oracle Net Services, 4-9
- directory usage configuration
 - after database installation, 5-3 to 5-4
 - client installation, 5-9
- distinguished names, 2-9
- distribution, of directories, 3-3
- DITs. *See* directory information trees
- DNs (distinguished names), 2-9

E

- enterprise domains, 4-12, 4-15
- enterprise roles, 4-11

G

- global roles, 4-11
- global topics, 4-20
- Globalization Support, 2-6

I

- Intelligent Client Failover, 3-4
- Intelligent Network Level Failover, 3-4
- interoperability, of Oracle products with third-party directories, 1-2

J

- Java Database Connectivity (JDBC)
 - Thin drivers
 - directory naming support, 4-4

K

- knowledge references. *See* referrals

L

- LDAP
 - benefits, 2-5
 - C API, 2-7
 - command-line tools
 - ldapadd, B-5
 - ldapbind, B-3
 - ldapdelete, B-6
 - ldapmoddn, B-8
 - ldapmodify, B-7
 - ldapsearch, B-4
 - definition, 2-5
 - extensibility features, 2-6
 - Globalization Support, 2-6
 - history, 2-5
 - purpose, 1-2
 - referrals, 2-6
 - schema discovery features, 2-6
 - security features, 2-6
 - version 3, 2-6
- LDAP version 3, 2-6
- ldapadd command-line tool, B-5
- ldapbind command-line tool, B-3
- ldapdelete command-line tool, B-6
- ldapmoddn command-line tool, B-8
- ldapmodify command-line tool, B-7
- ldapsearch command-line tool, B-4
- LDIF, 2-7
- LDIF files
 - description, 2-7
 - examples, 2-7
 - format, 2-7
 - modification, 2-7
 - types, 2-7
- Lightweight Directory Access Protocol. *See* LDAP

Lightweight Directory Interchange Format. *See* LDIF
load estimation, of directories, 3-5

M

mappings, user-to-schema, 4-12
metadirectory
 definition, 1-5
multimaster replication
 benefits, 3-3
 definition, 3-3

N

namespace design, 3-2, 3-3
naming contexts
 definition, 2-15
 publishing, 2-15
net service aliases, 4-5
net service names (database connect
 descriptors), 4-5

O

object classes
 abstract, 2-15
 Application Context, A-5
 auxiliary, 2-14, 2-15
 creation and redefinition, 2-15
 example, 2-13
 Oracle Advanced Queuing, A-6
 Oracle Advanced Security, A-4
 Oracle Dynamic Services, A-8, A-9
 Oracle Net Services, 4-8, A-2
 structural, 2-14
 subclasses, 2-15
 types, 2-14
online directories
 benefits, 2-2
 definition, 2-2
Oracle Advanced Queuing
 attributes, A-6, A-7
 directory deployment factors, 4-22
 directory entries, 4-21
 directory information tree, 4-21

 object classes, A-6
 overview, 1-3
 product summary, 4-20
 security measures, 4-22
Oracle Advanced Security
 ACLs, 4-14
 administrative groups, 4-14
 directory deployment factors, 4-15
 directory entries, 4-12
 directory information tree, 4-13
 object classes, A-4
 overview, 1-2
 product summary, 4-11
Oracle Context
 configuration, 2-19
 creation, 5-5
 definition, 2-19
 selection, 5-6, 5-8, 5-9
 structure, 2-19, 2-21
 upgrade, 5-5
 use of multiple contexts, 4-10
Oracle Directory Integration Platform
 features, 1-5
 purpose, 1-5
Oracle Dynamic Services
 attributes, A-9, A-10
 directory deployment factors, 4-30
 directory entries, 4-28
 directory information tree, 4-29
 object classes, A-8, A-9
 overview, 1-3
 product summary, 4-23
 security measures, 4-30
Oracle Internet Directory
 features, 1-4
 information flow, 2-8
 interoperability with third-party
 directories, 1-2, 1-5
 overview, 1-4
Oracle Names LDAP Proxy servers, 4-9, 4-10
Oracle Names servers, 4-9, 4-10
Oracle Net Configuration Assistant, 2-19, 5-3 to 5-4
 OracleDBC creators group, 4-9
Oracle Net Manager, 4-5, 4-9, 4-10

Oracle Net Services

- attributes, A-2, A-3
- configuration for directory naming, 4-9, 4-10
- database connectivity features, 4-2
- directory deployment factors, 4-10
- directory entries, 4-5
- directory information tree, 4-6
- directory naming, 4-3
- Internet scalability features, 4-2
- Internet security features, 4-2
- network management features, 4-2
- object classes, 4-8, A-2
- Oracle Net Manager, 4-5, 4-9, 4-10
- overview, 1-2
- security measures, 4-9

Oracle Schema

- creation, 5-6
- definition, 5-2
- upgrade, 5-6

OracleContextAdmins group, 4-9

OracleDBAdmins group, 4-14

OracleDBCreators group, 4-9, 4-14

OracleDBSecurityAdmins group, 4-14, 4-19

OracleDomainAdmins group, 4-14, 4-19

OraclePasswordAccessibleDomains group, 4-14

OracleUserSecurityAdmins group, 4-14

P

partitions, directory

- benefits, 3-4
- drawbacks, 3-4

R

RDNs (relative distinguished names), 2-9

referrals, 2-6

relative distinguished names, 2-9

replication, of directories, 3-3

- benefits, 3-3
- definition, 3-3

root DSE (Directory Server-Specific Entry), 2-6, 2-16

root Oracle Context, 5-5

S

schema, 2-16

schemas, shared, 4-12

Simple Authentication and Security Layer (SASL), 2-6

single password authentication, 4-12

single sign-on using a centrally stored wallet, 4-12

structural object classes, 2-14

subclasses, of object classes, 2-15

system requirements, of directories, 3-5

T

testing, of directories, 3-6

W

wallets, 4-12

X

X.500 protocol, 2-5

