Oracle® Internet Directory

Administrator’s Guide

Release 3.0.1

June 2001

Part No. A90151-01

Oracle Internet Directory Administrator’s Guide, Release 3.0.1

Part No. A90151-01

Primary Author: Richard Smith

Contributing Authors: Michael Cowan, Rebecca Reitmeyer, Ginger Tabora

Contributors: Tridip Bhattacharya, Saheli Dey, Bruce Ernst, Rajinder Gupta, Ajay Keni, Stephen Lee, David Lin, Michael Mesaros, Radhika Moolky, Hari Sastry, David Saslav, Daniele Schechter, Gurudat Shakshikumar, Amit Sharma, Daniel Shih, Saurabh Shrivastava, Uppili Srinivasan, Tsai Rung-Huang

Graphic Artist: Valarie Moore

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 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.

RSA and RC4 are trademarks of RSA Data Security. Portions of Oracle Internet Directory have been licensed by Oracle Corporation from RSA Data Security.

Oracle Directory Manager requires the JavaTM Runtime Environment. The JavaTM Runtime Environment, Version JRE 1.1.6. ("The Software") is developed by Sun Microsystems, Inc. 2550 Garcia Avenue, Mountain View, California 94043. Copyright (c) 1997 Sun Microsystems, Inc.

This product contains SSLPlus Integration Suite‘, version 1.2, from Consensus Development Corporation. Oracle Directory Manager requires the Java‘ Runtime Environment. The Java‘ Runtime Environment, Version JRE 1.1.6. ("The Software") is developed by Sun Microsystems, Inc. 2550 Garcia Avenue, Mountain View, California 94043. Copyright (c) 1997 Sun Microsystems, Inc.

Oracle is a registered trademark, and SQL*Net, SQL*Loader, SQL*Plus, Net8, Oracle Net Services, and

Contents

Part I Getting Started

1 Introduction

2 Concepts and Architecture

3 General Deployment Considerations

4 Preliminary Tasks

5 Using the Administration Tools

Part II Basic Directory Administration

6 Managing the Oracle Directory Server

7 Managing the Directory Schema

8 Managing Directory Entries

9 Managing Globalization Support in the Directory

10 Managing the Delegated Administration Service

Part III Directory Security

11 About Security in Oracle Internet Directory

12 Managing Secure Sockets Layer (SSL)

13 Managing Directory Access Control

Part IV Directory Replication

14 About Directory Replication

15 Managing Directory Replication

16 Adding a Node to a DRG by Using the Database Copy Procedure

Part V Directory Deployment

17 Capacity Planning Considerations

18 High Availability And Failover Considerations

19 Tuning Considerations

Part VI The Directory and Clusters

20 Managing Failover in Cluster Configurations

21 Managing Directory Failover in an Oracle9*i* Real Application Clusters Environment

Part VII The Oracle Directory Integration Platform

22 About the Oracle Directory Integration Platform

23 Managing Directory Integration Agents and Profiles

24 Managing the Oracle Directory Integration Server

26 Bootstrapping a Directory in the Oracle Directory Integration Platform

27 Synchronizing with Oracle Human Resources

Part VIII Appendixes

A Syntax for LDIF and Command-Line Tools

B Using Access Control Directive Format

C Schema Elements

D Using Oracle Wallet Manager

E Upgrading from Oracle Internet Directory Release 2.1.1

F Migrating Data from Other LDAP-Compliant Directories

G Troubleshooting

Glossary

Index

Send Us Your Comments

Oracle Internet Directory Administrator’s Guide, Release 3.0.1

Part No. A90151-01

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

Oracle Internet Directory Administrator's Guide describes the features, architecture, and administration of Oracle Internet Directory. For information about installation, see the installation documentation for your operating system.

This preface contains these topics:

- Audience
- Organization
- Related Documentation
- Conventions
- Documentation Accessibility

Audience

Oracle Internet Directory Administrator's Guide is intended for anyone who performs administration tasks for the Oracle Internet Directory. You should be familiar with either the UNIX operating system or the Microsoft Windows NT operating system in order to understand the line-mode commands and examples. You can perform all of the tasks through the line-mode commands, and you can perform most of the tasks through Oracle Directory Manager, which is operating system-independent.

To use this document, you need some familiarity with the **Lightweight Directory Access Protocol (LDAP)**.

Organization

This document contains the chapters and appendixes listed in this section. Oracle Corporation encourages you to read the conceptual and other introductory material presented in Part I before performing installation and maintenance.

Part I: Getting Started

Part I provides an overview of the product and its features, a conceptual foundation necessary to configure and manage a directory.

Chapter 1, "Introduction"

This chapter provides an introduction to directories, LDAP, and Oracle Internet Directory features.

Chapter 2, "Concepts and Architecture"

This chapter gives an overview of online directories and Lightweight Directory Access Protocol (LDAP). Provides conceptual descriptions of directory entries, attributes, object classes, naming contexts, schemas, distributed directories, security, and National Language Support. It also discusses Oracle Internet Directory architecture.

Chapter 3, "General Deployment Considerations"

This chapter discusses general issues to consider when deploying Oracle Internet Directory. This chapter helps you assess the requirements of a directory in an enterprise and make effective deployment choices.

Chapter 4, "Preliminary Tasks"

This chapter discusses how to prepare your directory for configuration and use. It tells you how to start and stop OID Monitor and instances of Oracle directory server and Oracle directory replication server. It discusses the need to reset the default security configuration, how to upgrade from earlier releases of Oracle Internet Directory, and how to migrate data from other LDAP-compliant directories.

Chapter 5, "Using the Administration Tools"

This chapter explains how to use the various administration tools: Oracle Directory Manager, command-line tools, bulk tools, Catalog Management tool, OID Database Password Utility, replication tools, and Database Statistics Collection tool

Part II: Basic Directory Administration

Part II guides you through the tasks required to configure and maintain Oracle Internet Directory.

Chapter 6, "Managing the Oracle Directory Server"

This chapter provides instructions for managing server configuration set entries; setting system operational attributes; managing naming contexts and password encryption; configuring searches; managing super, guest, and proxy users; setting debug logging levels; using audit log; viewing active server instance information; and changing the password to an Oracle database server.

Chapter 7, "Managing the Directory Schema"

This chapter explains what a directory schema is, what an object class is, and what an attribute is. It tells you how to manage the Oracle Internet Directory schema by using Oracle Directory Manager and the command-line tools.

Chapter 8, "Managing Directory Entries"

This chapter explains how to search, view, add, modify and manage entries by using Oracle Directory Manager and the command-line tools.

Chapter 9, "Managing Globalization Support in the Directory"

Discusses National Language Support (NLS) as used by Oracle Internet Directory.

Chapter 10, "Managing the Delegated Administration Service"

This chapter explains the Delegated Administration Service, which enables directory users to modify their own personal data—such as addresses, phone numbers, and photos—without the intervention of an administrator. It also enables users to search other parts of the directory to which they have access. This frees directory administrators for other tasks in the enterprise.

Part III Directory Security

Part III tells how to secure data within the directory itself and within an enterprise deployment of a directory.

Chapter 11, "About Security in Oracle Internet Directory"

This chapter describes the security features available with Oracle Internet Directory, and explains how to deploy the directory for administrative delegation.

Chapter 12, "Managing Secure Sockets Layer (SSL)"

This chapter introduces and explains how to configure the features of Secure Sockets Layer (SSL).

Chapter 13, "Managing Directory Access Control"

This chapter provides an overview of access control policies and describes how to administer directory access.

Part IV Directory Replication

Part IV provides a detailed discussion of replication and how to manage it.

Chapter 14, "About Directory Replication"

This chapter expands on the discussion about replication in Chapter 2, "Concepts and Architecture".

Chapter 15, "Managing Directory Replication"

This chapter explains how to install and initialize Oracle directory replication server software the first time, and how to install new nodes into an environment where that software is already installed.

Chapter 16, "Adding a Node to a DRG by Using the Database Copy Procedure"

This chapter describes an alternate method of adding a node to a replicated directory system if the directory is very large.

Part V: Directory Deployment

Part V discusses important deployment considerations, including capacity planning, high availability, and tuning.

Chapter 17, "Capacity Planning Considerations"

This chapter tells you how to assess applications' directory access requirements and ensure that the Oracle Internet Directory has adequate computer resources to service requests at an acceptable rate.

Chapter 18, "High Availability And Failover Considerations"

This chapter describes the availability and failover features of various components in the Oracle Internet Directory technology stack, and provides guidelines for exploiting them optimally for typical directory deployment.

Chapter 19, "Tuning Considerations"

This chapter gives guidelines for ensuring that the combined hardware and software are yielding the desired levels of performance.

Part VI: Oracle Internet Directory and Clusters

Part VI discusses cluster support in Oracle Internet Directory.

Chapter 20, "Managing Failover in Cluster Configurations"

This chapter explains how to increase high availability by using logical hosts—as opposed to physical hosts—in clustered environments.

Chapter 21, "Managing Directory Failover in an Oracle9i Real Application Clusters Environment"

This chapter discusses the ways you can run Oracle Internet Directory in an Oracle Real Application Clusters system.

Part VII: The Oracle Directory Integration Platform

Part VII explains the concepts, architecture, and components of the Oracle Directory Integration platform, and tells you how to configure and use it to synchronize multiple directories with Oracle Internet Directory.

Chapter 22, "About the Oracle Directory Integration Platform"

This chapter introduces the Oracle Directory Integration platform, its components, architecture, and administration tools.

Chapter 23, "Managing Directory Integration Agents and Profiles"

This chapter discusses directory integration agents and the operations they perform in the Oracle Directory Integration platform. It explains how to manage partner agents by using either Oracle Directory Manager of command-line tools.

Chapter 24, "Managing the Oracle Directory Integration Server"

This chapter discusses the Oracle directory integration server and tells you how to configure and manage it.

Chapter 25, "Managing Security in the Oracle Directory Integration Platform"

This chapter discusses the most important aspects of security in the Oracle Directory Integration platform.

Chapter 26, "Bootstrapping a Directory in the Oracle Directory Integration Platform"

This chapter explains some of the initial setup tasks you may need to perform as you begin using the Oracle Directory Integration platform.

Chapter 27, "Synchronizing with Oracle Human Resources"

If you store employee data in Oracle Internet Directory, and if you use Oracle Human Resources to create, modify, and delete that data, then you must ensure that the data is synchronized between the two. This chapter explains the Oracle Human Resources agent, which enables you to do this.

Part VIII: Appendixes

Appendix A, "Syntax for LDIF and Command-Line Tools"

This appendix provides syntax, usage notes, and examples for LDAP Data Interchange Format and LDAP command-line tools.

Appendix B, "Using Access Control Directive Format"

This appendix describes the format (syntax) of Access Control Information Items(ACIs).

Appendix C, "Schema Elements"

This appendix lists schema elements supported in Oracle Internet Directory.

Appendix D, "Using Oracle Wallet Manager"

This appendix describes and explains how to use Oracle Wallet Manager to create and manage wallets and certificates.

Appendix E, "Upgrading from Oracle Internet Directory Release 2.1.1"

This appendix tells you how to upgrade to Oracle Internet Directory release 3.0.1 from Oracle Internet Directory release 2.1.1.

Appendix F, "Migrating Data from Other LDAP-Compliant Directories"

This appendix explains the steps to migrate data from LDAP v3-compatible directories into Oracle Internet Directory.

Appendix G, "Troubleshooting"

This appendix lists possible failures and error codes and their probable causes.

Related Documentation

For more information, see:

- Online help available through Oracle Directory Manager
- The Oracle9*i* documentation set, especially:
  - *Oracle9i Database Administrator's Guide*
  - *Oracle Directory Service Integration and Deployment Guide*
  - *Oracle Internet Directory Application Developer's Guide*
  - *Oracle Net Services Administrator's Guide*
  - *Oracle9i Real Application Clusters Administration*
  - *Oracle9i Replication*

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://technet.oracle.com/membership/index.htm`

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://technet.oracle.com/docs/index.htm`

For additional information, see:

- Chadwick, David. *Understanding X.500—The Directory*. Thomson Computer Press, 1996.
- Howes, Tim and Mark Smith. *LDAP: Programming Directory-enabled Applications with Lightweight Directory Access Protocol*. Macmillan Technical Publishing, 1997.
- Howes, Tim, Mark Smith and Gordon Good, *Understanding and Deploying LDAP Directory Services*. Macmillan Technical Publishing, 1999.
- Internet Assigned Numbers Authority home page, `http://www.iana.org`, for information about object identifiers
- Internet Engineering Task Force (IETF) documentation, especially:
  - `http://www.ietf.org` for the IETF home page
  - `http://www.ietf.org/html.charters/ldapext-charter.html` for the ldapext charter and LDAP drafts)
  - `http://www.ietf.org/html.charters/ldup-charter.html` for the LDUP charter and drafts
  - `http://www.ietf.org/rfc/rfc2254.txt`, "The String Representation of LDAP Search Filters"
  - `http://www.ietf.org/rfc/rfc1823.txt`, "The LDAP Application Program Interface"
- The OpenLDAP Community, `http://www.openldap.org`

Conventions

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

Conventions in Text

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

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle9i Database Concepts* Ensure that the recovery catalog and target database do *not* reside on the same disk.
UPPERCASE monospace (fixed-width font)	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, user names, 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, user names 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.

Convention	Meaning	Example
lowercase monospace (fixed-width font) italic	Lowercase monospace italic font represents placeholders or variables.	You can specify the *parallel_clause*. Run U*old_release*.SQL where *old_release* refers to the release you installed prior to upgrading.

Conventions in Code Examples

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

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

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

Convention	Meaning	Example
[ ]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (*digits* [, *precision*])
{ }	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:
	■ 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 *subquery*;
		SELECT *col1*, *col2*, ... , *coln* FROM employees;

. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	

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

`Italics`	Italicized text indicates placeholders or variables for which you must supply particular values.	`CONNECT SYSTEM/`*`system_password`*
		`DB_NAME = `*`database_name`*

`UPPERCASE`	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. 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. For example, to start Oracle Database Configuration Assistant, you must click the Start button on the taskbar and then choose Programs > Oracle - *HOME_NAME* > Database Administration > Database Configuration Assistant.	Choose Start > Programs > Oracle - *HOME_NAME* > Database Administration > Database Configuration Assistant
`C:\>`	Represents the Windows command prompt of the current hard disk drive. Your prompt reflects the subdirectory in which you are working. Referred to as the command prompt in this guide.	`C:\oracle\oradata>`
HOME_NAME	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	`C:\> net start Oracle`*`HOME_NAME`*`TNSListener`

Convention	Meaning	Example
ORACLE_HOME and *ORACLE_BASE*	In releases prior to 8.1, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory that by default was: - C:\orant for Windows NT - C:\orawin95 for Windows 95 - C:\orawin98 for Windows 98 or whatever you called your Oracle home. In this Optimal Flexible Architecture (OFA)-compliant release, all subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is C:\oracle. If you install release 9.0 on a computer with no other Oracle software installed, the default setting for the first Oracle home directory is C:\oracle\ora90. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. See *Oracle9i Database Getting Starting for Windows* for additional information on OFA compliances and for information on installing Oracle products in non-OFA compliant directories.	Go to the *ORACLE_BASE*\ORACLE_HOME\rdbms\admin directory.

Documentation Accessibility

Oracle's goal is to make our products, services, and supporting documentation accessible to the disabled community with good usability. 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 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/`

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.

What’s New in Oracle Internet Directory?

This section provides a brief description of new features introduced with the latest releases of Oracle Internet Directory, and points you to more information about each one. It contains these topics:

- New Features Introduced with Oracle Internet Directory Release 3.0.1
- New Features Introduced with Oracle Internet Directory Release 2.1.1

New Features Introduced with Oracle Internet Directory Release 3.0.1

This section describes the new features introduced with Oracle Internet Directory release 3.0.1.

- **Capability to run multiple Oracle Internet Directory instances on the same host**

  This new feature enables you to run more than one installation of Oracle Internet Directory on a single host. You can then replicate between them or use this new feature as part of a failover strategy.

  > **See Also:** "Running Multiple Installations of Oracle Internet Directory on One Host" on page 3-12

- **Delegated Administration Service**

  This new service enables directory users to modify their own personal data—such as addresses, phone numbers, and photos—without the intervention of an administrator. It also enables users to search other parts of the directory to which they have access. This frees directory administrators for other tasks in the enterprise.

  > **See Also:** Chapter 10, "Managing the Delegated Administration Service"

- **Oracle Directory Integration platform**

  This new feature enables you to synchronize various directories with Oracle Internet Directory. It also makes it easier for third party metadirectory vendors and developers to develop and deploy their own connectivity agents.

  > **See Also:** Part VII: "The Oracle Directory Integration Platform"

- **Password policy management**

  Password policy management enables you to establish and enforce rules for how passwords are used.

  > **See Also:**
  >
  > - "Password Policies" for a conceptual discussion
  > - "Managing Password Policies" on page 6-17

- **Upgrade procedures**

  These procedures enable you to upgrade from Oracle Internet Directory release 2.1.1.

  > **See Also:** Appendix E, "Upgrading from Oracle Internet Directory Release 2.1.1"

New Features Introduced with Oracle Internet Directory Release 2.1.1

This section describes the new features introduced with Oracle Internet Directory release 2.1.1.

- **Attribute options, including language codes**

  Attribute options enable you to specify how the value for an attribute is made available in a search or a compare operation. For example, suppose that an employee has two addresses, one in London, the other in New York. Options for that employee's `address` attribute could allow you to store both addresses. Users could then search for either address.

  Attribute options can include language codes. For example, options for John Doe's `givenName` attribute could enable you to store his given name in both French and Japanese. A user could then search for the name in either language.

  > **See Also:**
  >
  > - "Attribute Options" on page 2-7 for a conceptual discussion
  > - "Managing Entries with Attribute Options by Using Oracle Directory Manager" on page 8-13
  > - "Managing Entries with Attribute Options by Using Command-Line Tools" on page 8-17

- **Change log purging enhancements**

  These enhancements enable you to specify the type of change log purging to use: change number-based or time-based.

  > **See Also:**
  >
  > - "Change Log Purging" on page 14-6 for a conceptual discussion
  > - "Directory Replication Server Parameters" on page 15-11

- **Enhanced support for these operational attributes:**

  - **creatorsName**
  - **createTimestamp**
  - **modifiersName**
  - **modifyTimestamp**

  This enhanced support enables you to use one or more of these attributes in searches.

  See Also:

  - "Kinds of Attribute Information" on page 2-5 for a conceptual discussion
  - "Example 7: Searching for All User Attributes and Specified Operational Attributes" on page A-25 for an example of a search operation using the `createTimestamp` attribute

- **Migration from other LDAP-compliant directories**

  This new feature enables you to migrate data from other LDAP v3-compatible directories into Oracle Internet Directory.

  See Also: Appendix F, "Migrating Data from Other LDAP-Compliant Directories"

- **Object class explosion**

  Object class explosion enables you to add or perform an operation on an entry without specifying the entire hierarchy of superclasses associated with that entry.

  See Also: ""Guidelines for Adding Object Classes" on page 7-3 for an explanation of how to use this feature when adding object classes

- **OID database statistics collection tool**

  This tool assists in capacity planning. It helps you analyze the various database schema objects so that you can estimate the statistics.

  See Also: "Using the OID Database Statistics Collection Tool" on page 5-15

- **Password protection enhancements**

  This new feature enhances the available password protection by storing passwords as hashed values. Storing passwords as one-way hashed values—rather than as encrypted values—more fully secures them because a malicious user can neither read nor decrypt them. You can select one of the following hashing algorithms:

  - **MD4**—A one-way hash function that produces a 128-bit hash
  - **MD5**—An improved, and more complex, version of MD4
  - **SHA**—Secure Hash Algorithm, which produces a 160-bit hash, longer than MD5. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.
  - **UNIX Crypt**—The UNIX encryption algorithm
  - No Encryption

    See Also:

    - "Password Protection" on page 11-6 for a conceptual discussion
    - "Managing Password Protection" on page 6-21 for instructions on setting password encryption

- **Replication tools**

  The following new replication tools are now added:

  - **Human intervention queue manipulation tool**

    This tool enables you to move changes from the human intervention queue to either the retry queue or the purge queue.

  - **OID reconciliation tool**

    This tool enables you to synchronize conflicting changes in a replicated environment.

See Also:

- "Using the Replication Tools" on page 5-15 for a brief explanation of this tool
- "Using the Human Intervention Queue Manipulation Tool" on page 15-31
- "Using the OID Reconciliation Tool" on page 15-32

- **Replication node deletion**

  This new feature enables you to delete a node from a directory replication group.

  See Also: "Deleting a Replication Node" on page 15-26

- **Synchronization with multiple directories in a metadirectory environment (release 2.1.1 only)**

  If you are working in a metadirectory environment, then this new feature enables you to form a single virtual directory by synchronizing multiple directories with Oracle Internet Directory.

  Note: This feature was replaced in release 3.0.1 by the Oracle Directory Integration platform. See Chapter 22, "About the Oracle Directory Integration Platform" for further information.

- **Upgrade procedures (release 2.1.1 only)**

  These new procedures enable you to upgrade from either Oracle Internet Directory release 2.0.4.x or release 2.0.6. Not supported in release 2.1.1.1 or in release 3.0.1.

  See Also: Appendix E, "Upgrading from Oracle Internet Directory Release 2.1.1"

Part I

Getting Started

Part I explains what Oracle Internet Directory is and some of the concepts you must know before using it. It contains these chapters:

- Chapter 1, "Introduction"
- Chapter 2, "Concepts and Architecture"
- Chapter 3, "General Deployment Considerations"
- Chapter 4, "Preliminary Tasks"
- Chapter 5, "Using the Administration Tools"

1

Introduction

This chapter describes some of the information management challenges your enterprise faces in the Internet age, and how Oracle Internet Directory helps you meet them.

This chapter contains these topics:

- What Is a Directory?
- What Is LDAP?
- What Is Oracle Internet Directory?

What Is a Directory?

Directories organize complex information so that we can find it easily. They list objects—for example, people, books in a library, or merchandise in a department store—and give details about each one. You probably use several offline directories everyday: a telephone book, a card catalog in a library, or a department store catalog, to mention a few.

Enterprises with distributed computer systems use *online* directories for fast searches, cost-effective management of users and security, and a central integration point for multiple applications and services. Online directories are also becoming critical to both e-businesses and hosted environments.

This section contains these topics:

- The Expanding Role of Online Directories
- The Problem: Too Many Special Purpose Directories

The Expanding Role of Online Directories

An online directory is a specialized database that stores and retrieves collections of information about objects. Such information can represent any resources that require management: employee names, titles, and security credentials; information about partners; or information about shared network resources such as conference rooms and printers.

Online directories can be used by a variety of users and applications, and for a variety of purposes, including:

- An employee searching for corporate whitepage information, and, through a mail client, looking up email addresses
- An application, such as a message transport agent, locating a user’s mail server
- A database application identifying user role information

Although an online directory is a database—that is, a structured collection of data—it is not a **relational database**. The following table contrasts online directories with relational databases.

Online Directories	Relational Databases
Primarily read-focused. Typical use involves a relatively small number of data updates, and a potentially large number of data retrievals.	**Primarily write-focused.** Typical use involves continuous recording of transactions, with retrievals done relatively infrequently.
Designed to handle relatively simple transactions on relatively small units of data. For example, an application might use a directory simply to store and retrieve an e-mail address, a telephone number, or a digital portrait.	**Designed to handle large and diverse transactions using many operations on large units of data.**
Designed to be location-independent. 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.	**Typically designed to be location-specific.** While a relational database can be distributed, it usually resides on a particular database server.
Designed to store information in entries. These entries might represent any resource customers wish to manage: employees, e-commerce partners, conference rooms, or shared network resources such as printers. Associated with each entry is a number of attributes, each of which may have one or more values assigned. For example, typical attributes for a `person` entry might include first and last names, e-mail addresses, the address of a preferred mail server, passwords or other login credentials, or a digitized portrait.	**Designed to store information as rows in relational tables.**

The Problem: Too Many Special Purpose Directories

According to some estimates, each of the world's largest companies has an average of 180 different directories, each designated for a special purpose. Add to this the various enterprise applications, each with its own additional directory of user names, and the actual number of special purpose directories becomes even greater.

Managing so many special purpose directories can cause problems:

- High cost of administration: Administrators must maintain essentially the same information in many different places. For example, when an enterprise hires a new employee, administrators must create a new user identity on the network, create a new e-mail account, add the user to the human-resources database, and set up all applications that the employee may need—for example, user accounts on development, testing, and production database systems. Later, if the employee leaves the company, administrators must reverse the process to disable all these user accounts.
- Inconsistent data: Because of the large administrative overhead, it can be difficult for multiple administrators, entering redundant information in multiple systems, to synchronize this employee information across all systems. The result can be inconsistent data across the enterprise.

Clearly there is need for a more general purpose directory infrastructure, one based on a common standard for supporting a wide variety of applications and services.

What Is LDAP?

LDAP is a standard, extensible directory access protocol. It is a common language that LDAP clients and servers use to communicate.

This section contains these topics:

- LDAP and Simplified Directory Management
- LDAP Version 3

LDAP and Simplified Directory Management

LDAP was conceived as an Internet-ready, lightweight implementation of the International Standardization Organization (ISO) X.500 standard for directory services. It requires a minimal amount of networking software on the client side, which makes it particularly attractive for Internet-based, thin client applications.

The LDAP standard simplifies management of directory information in three ways:

- It provides all users and applications in the enterprise with a single, well-defined, standard interface to a single, extensible directory service. This makes it easier to rapidly develop and deploy directory-enabled applications.
- It reduces the need to enter and coordinate redundant information in multiple services scattered across the enterprise.
- 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 LDAP, Version 3, was approved as a proposed Internet Standard by the **Internet Engineering Task Force (IETF)** in December 1997. LDAP Version 3 improves on LDAP Version 2 in several important areas:

- Globalization Support: LDAP Version 3 allows servers and clients to support characters used in every language in the world.
- Knowledge references (also called referrals): LDAP Version 3 implements a referral mechanism that allows servers to return references to other servers as a result of a directory query. This makes it possible to distribute directories globally by partitioning a **directory information tree (DIT)** across multiple LDAP servers.
- Security: LDAP Version 3 adds a standard mechanism for supporting **Simple Authentication and Security Layer (SASL)** and **Transport Layer Security (TLS)**, providing a comprehensive and extensible framework for data security.
- Extensibility: LDAP Version 3 enables vendors to extend existing LDAP operations through the use of mechanisms called controls.
- Feature and schema discovery: LDAP Version 3 enables publishing information useful to other LDAP servers and clients, such as the supported LDAP protocols and a description of the directory schema.

See Also:

- RFCs (Requests for Comments) 2251-2256 of the IETF, available on the Worldwide Web at: http://www.ietf.org/rfc.html
- "Related Documentation" on page xxxvii for an additional list of resources on LDAP
- Chapter 2, "Concepts and Architecture" for a conceptual discussion of directory information trees and knowledge references

What Is Oracle Internet Directory?

Oracle Internet Directory is a general purpose directory service that enables fast retrieval and centralized management of information about dispersed users and network resources. It combines **Lightweight Directory Access Protocol (LDAP)** Version 3 with the high performance, scalability, robustness, and availability of Oracle9*i*.

This section contains these topics:

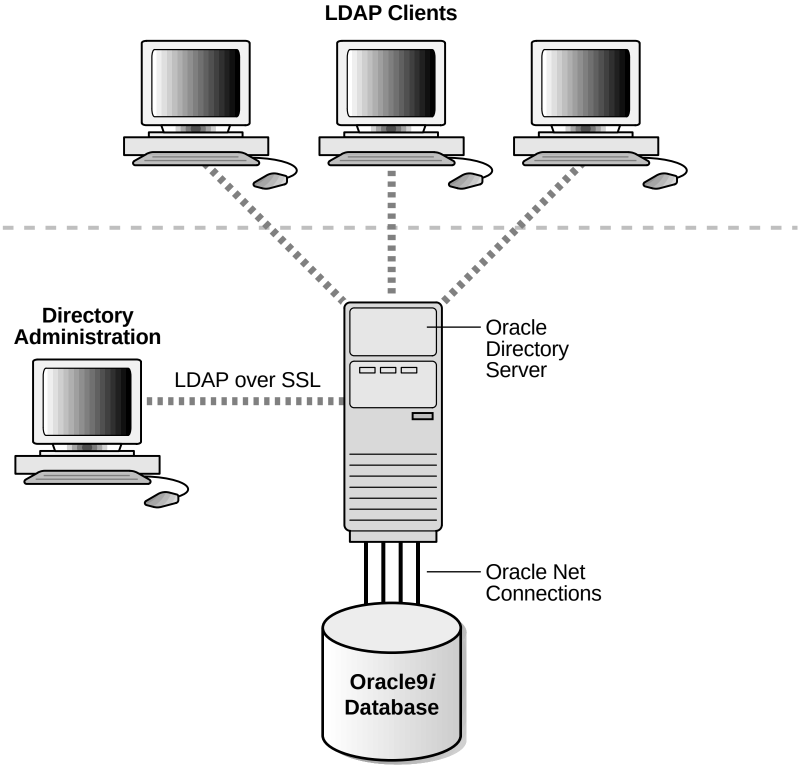
- Oracle Internet Directory Architecture
- Oracle Internet Directory Components
- The Advantages of Oracle Internet Directory

Oracle Internet Directory Architecture

Oracle Internet Directory runs as an application on Oracle9*i*. It communicates with the database, which may or may not be on the same operating system, by using

Oracle Net Services, Oracle's operating system-independent database connectivity solution. Figure 1–1 illustrates this relationship.

Figure 1–1 Oracle Internet Directory Architecture



Oracle Internet Directory Components

Oracle Internet Directory includes:

- Oracle directory server, which responds to client requests for information about people and resources, and to updates of that information, by using a multitiered architecture directly over TCP/IP
- Oracle directory replication server, which replicates LDAP data between Oracle directory servers

- Directory Administration, which includes:
  - Oracle Directory Manager, a Java-based graphical user interface administration tool, which simplifies directory administration
  - A variety of command-line administration and data management tools invoked from LDAP clients
- Oracle Directory Integration platform, including the Oracle directory integration server, which enables you to synchronize various directories with Oracle Internet Directory. You can also use the Oracle Directory Integration platform to develop and deploy your own connectivity agents.
- The Delegated Administration Service, which enables directory users to modify their own personal data—such as addresses, phone numbers, and photos—without the intervention of an administrator

The Advantages of Oracle Internet Directory

Among its more significant benefits, Oracle Internet Directory provides scalability, high availability, and security.

Scalability

Oracle Internet Directory exploits the strengths of Oracle9*i*, enabling support for terabytes of directory information. In addition, such technologies as multithreaded LDAP servers and database connection pooling allow it to support thousands of concurrent clients with subsecond search response times.

Oracle Internet Directory also provides data management tools, such as Oracle Directory Manager and a variety of command-line tools, for manipulating large volumes of LDAP data.

High Availability

Oracle Internet Directory is designed to meet the needs of a variety of important applications. For example, it supports full, multimaster replication between directory servers: If one server in a replication community becomes unavailable, then a user can access the data from another server. Information about changes made to directory data on a server is stored in special tables on the Oracle9*i* database. These are replicated throughout the directory environment by **Oracle9i Replication**, a robust replication mechanism.

Oracle Internet Directory also takes advantage of all the availability features of the Oracle9*i*. Because directory information is stored securely in the Oracle9*i* database, it is protected by Oracle’s backup capabilities. Additionally, the Oracle9*i* database, running with large datastores and heavy loads, can recover from system failures quickly.

Security

Oracle Internet Directory offers comprehensive and flexible access control. An administrator can grant or restrict access to a specific directory object or to an entire directory subtree. Moreover, Oracle Internet Directory implements three levels of user authentication: anonymous, password-based, and certificate-based using **Secure Socket Layer (SSL)** Version 3 for authenticated access and data privacy.

2

Concepts and Architecture

This chapter provides conceptual descriptions of the basic elements of Oracle Internet Directory and discusses Oracle Internet Directory architecture.

This chapter contains these topics:

- Entries
- Attributes
- Object Classes
- Naming Contexts
- The Directory Schema
- Security
- Globalization Support
- Oracle Internet Directory Architecture
- Example: How Oracle Internet Directory Works
- Distributed Directories
- The Delegated Administration Service
- The Oracle Directory Integration Platform

> **See Also:** "Related Documentation" on page xxxvii for suggestions on further reading about LDAP-compliant directories

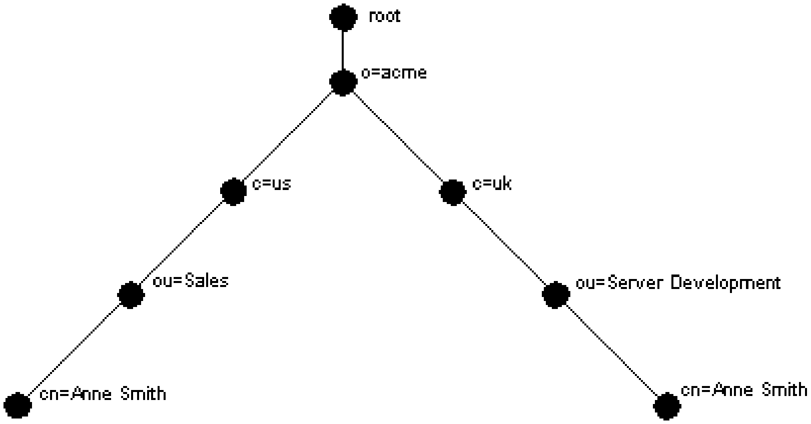
Entries

In a directory, each collection of information about an object is called an **entry**. For example, a typical telephone directory includes entries for people, and a library card catalog contains entries for books. Similarly, an online directory might include entries for employees, conference rooms, e-commerce partners, or shared network resources such as printers.

Each entry in a directory is uniquely identified by a **distinguished name (DN)**. The distinguished name tells you exactly where the entry resides in the directory's hierarchy. This hierarchy is represented by a **directory information tree (DIT)**.

To understand the relation between a distinguished name and a directory information tree, look at Figure 2–1.

Figure 2–1 A Directory Information Tree



The DIT in Figure 2–1 diagrammatically represents entries for two employees of Acme Corporation who are both named Anne Smith. It is structured along geographical and organizational lines. The Anne Smith represented by the left branch works in the Sales division in the United States, while the other works in the Server Development division in the United Kingdom.

The Anne Smith represented by the right branch has the common name (`cn`) Anne Smith. She works in an organizational unit (`ou`) named Server Development, in the country (`c`) of Great Britain (`uk`), in the organization (`o`) Acme.

The DN for this "Anne Smith" entry is:

```
cn=Anne Smith,ou=Server Development,c=uk,o=acme
```

Note that the conventional format of a distinguished name places the lowest DIT component at the left, then follows it with the next highest component, thus moving progressively up to the root.

Within a distinguished name, the lowest component is called the **relative distinguished name (RDN)**. For example, in the above entry for Anne Smith, the RDN is `cn=Anne Smith`. Similarly, the RDN for the entry immediately above Anne Smith's RDN is `ou=Server Development`, the RDN for the entry immediately above `ou=Server Development` is `c=uk`, and so on. A DN is thus a sequence of RDNs separated by commas.

To locate a particular entry within the overall DIT, a client uniquely identifies that entry by using the full DN—not simply the RDN—of that entry. For example, within the global organization in Figure 2–1, to avoid confusion between the two Anne Smiths, you would use each one's full DN. (If there are potentially two employees with the same name in the same organizational unit, you could use additional mechanisms, such as identifying each employee with a unique identification number.)

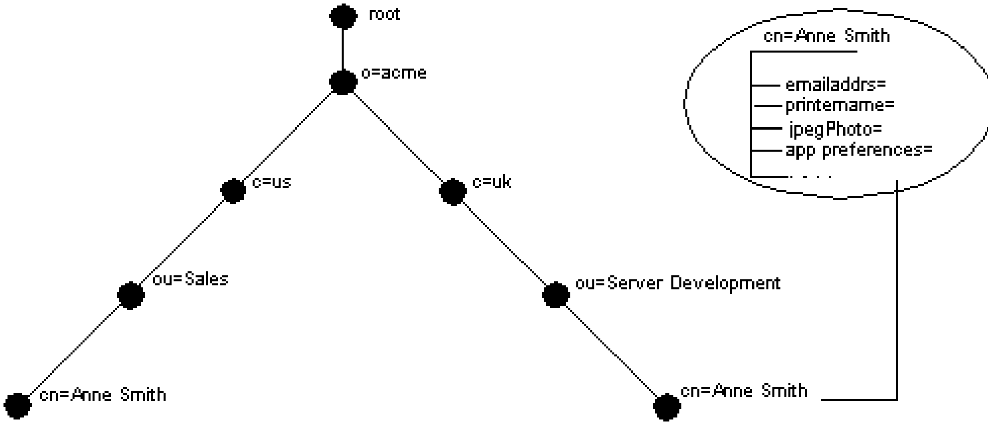
See Also: Chapter 8, "Managing Directory Entries."

Attributes

In a typical telephone directory, an **entry** for a person contains such information items as an address and a phone number. In an online directory, such an information item is called an **attribute**. Attributes in a typical employee entry can include, for example, a job title, an e-mail address, or a phone number.

For example, in Figure 2–2, the entry for Anne Smith in Great Britain (uk) has several attributes, each providing specific information about her. These are listed in the balloon to the right of the tree, and they include `emailaddrs`, `printername`, `jpegPhoto`, and `app preferences`. Moreover, each bullet in Figure 2–2 is also an entry with attributes, although the attributes for each are not shown.

Figure 2–2 Attributes of the Entry for Anne Smith



Each attribute consists of an attribute type and one or more attribute values. The **attribute type** is the kind of information that the attribute contains—for example, `jobTitle`. The **attribute value** is the particular occurrence of information appearing in that entry. For example, the value for the `jobTitle` attribute could be `manager`.

This section contains these topics:

- Kinds of Attribute Information
- Single-Valued and Multivalued Attributes
- Attribute Options
- Common LDAP Attributes
- Attribute Syntax
- Attribute Matching Rules

Kinds of Attribute Information

Attributes contain two kinds of information.

Application Information — This information is maintained and retrieved by the directory clients and is unimportant to the operation of the directory. A telephone number, for example, is application information.

Operational Information — This information pertains to the operation of the directory itself. Some operational information is specified by the directory to control the server—for example, the time stamp for the creation or modification of an entry, or the name of the user who creates or modifies an entry. Other operational information, such as access information, is defined by administrators and is used by the directory program in its processing.

Any given attribute can hold either application information, or operational information, but not both.

To enhance your ability to search for entries, Oracle Internet Directory automatically creates several system operational attributes when you add an entry to the directory. These include:

`creatorsName`	Name of the person creating the entry
`createTimestamp`	Time of entry creation in **UTC (Coordinated Universal Time)**
`modifiersName`	Name of person creating the entry
`modifyTimestamp`	Time of entry creation in UTC

Moreover, when a user modifies an entry, Oracle Internet Directory automatically updates the `modifiersName` and `modifyTimestamp` attributes to, respectively, the name of the person modifying the entry, and the time of the entry modification in UTC.

> **See Also:** "Setting System Operational Attributes" on page 6-13 for instructions on configuring system operational attributes

Single-Valued and Multivalued Attributes

Attributes can be either single-valued or multivalued. Single-valued attributes carry only one value in the attribute, whereas multivalued attributes can have several. An example of a multivalued attribute is a group membership list with names of everyone in the group.

Common LDAP Attributes

Oracle Internet Directory implements all of the standard LDAP attributes. Table 2–1 shows some of the more common LDAP attributes.

Table 2–1 Common LDAP Attributes

Attribute Type	Attribute String	Description
commonName	`cn`	Common name of an entry—for example, `Anne Smith`
domainComponent	`dc`	The DN of the component in a Domain Name System (DNS)—for example, `dc=uk,dc=acme,dc=com`
jpegPhoto	`jpegPhoto`	Photographic image in JPEG format. The path and file name of the JPEG image you want to include as an entry attribute—for example, `/photo/audrey.jpg`
organization	`o`	Name of an organization—for example, `my_company`.
organizationalUnitName	`ou`	Name of a unit within an organization—for example, `Server Development`
owner	`owner`	Distinguished name of the person who owns the entry, for example, `cn=Anne Smith, ou=Server Development, o= Acme, c=uk`
surname, sn	`sn`	Last name of a person—for example, `Smith`
telephoneNumber	`telephoneNumber`	Telephone number—for example, `(650) 123-4567` or `6501234567`

See Also: Appendix C for a list of several proprietary attributes Oracle Internet Directory provides.

Attribute Syntax

Attribute syntax is the format of the data that can be loaded into each attribute. For example, the syntax of the `telephoneNumber` attribute might require a telephone number to be a string of numbers containing spaces and hyphens. However, the syntax for another attribute might require specifying whether the data has to be in the form of a date, or whether the data can consist of numbers only. Each attribute must have one and only one syntax attached to it.

Oracle Internet Directory recognizes most of the syntax specified in RFC 2252, allowing you to associate most of the syntax described in that document with an attribute. In addition to recognizing the syntax in RFC 2252, Oracle Internet Directory also enforces some LDAP syntax. You cannot add new syntaxes beyond those already supported by Oracle Internet Directory.

See Also: "LDAP Syntax" on page C-7

Attribute Matching Rules

In response to most incoming client requests, the directory server performs search and compare operations. During these operations, the directory server consults the relevant **matching rule** to determine equality between the attribute value sought and the attribute value stored. For example, matching rules associated with the `telephoneNumber` attribute could cause "(650) 123-4567" to be matched with either "(650) 123-4567" or "6501234567" or both. When you create an attribute, you associate a matching rule with it.

Oracle Internet Directory implements all the standard LDAP matching rules. You cannot add new matching rules beyond those already supported by Oracle Internet Directory.

See Also: "Matching Rules" on page C-10

Attribute Options

An attribute type can have various options that enable you to specify how the value for that attribute is made available in a search or a compare operation. For example, suppose that an employee has two addresses, one in London, the other in New York. Options for that employee’s `address` attribute could allow you to store both addresses.

Moreover, attribute options can include language codes. For example, options for John Doe’s `givenName` attribute could enable you to store his given name in both French and Japanese.

For clarity, we can distinguish between an attribute with an option and its base attribute, which is the same attribute without an option. For example, in the case of `cn;lang-fr=Jean`, the base attribute is `cn`; the French value for that base attribute is `cn;lang-fr=Jean`.

An attribute with one or more options inherits the properties—for example, matching rules and syntax— of its base attribute. To continue the previous example, the attribute with the option `cn;lang-fr=Jean` inherits the properties of `cn`.

Note: You cannot use an attribute option within a DN. For example, the following DN is incorrect: `cn;lang-fr=Jean, ou=sales,o=acme,c=uk`.

See Also:

- "Managing Entries with Attribute Options by Using Oracle Directory Manager" on page 8-13
- "Managing Entries with Attribute Options by Using Command-Line Tools" on page 8-17

Object Classes

An **object class** is a group of attributes that define the structure of an entry. When you define a directory **entry**, you assign one or more object classes to it. Some of the attributes in these object classes are mandatory, others are optional.

For example, the `organizationalPerson` object class includes the mandatory attributes `commonName` (`cn`) and `surname` (`sn`), and the optional attributes `telephoneNumber`, `uid`, `streetAddress`, and `userPassword`. When you define an entry by using the `organizationalPerson` object class, you must specify values for `commonName` (`cn`) and `surname` (`sn`). You do not need to provide values for `telephoneNumber`, `uid`, `streetAddress`, and `userPassword`.

At installation, Oracle Internet Directory provides standard LDAP object classes, as well as several proprietary object classes. You cannot add mandatory attributes to the sets of attributes belonging to these predefined object classes. If a given object class does not contain all the attributes that you want for an entry, then you can do one of the following:

- Add optional attributes to an existing object class
- Define a new (base) object class
- Define an object subclass

See Also: Appendix C for a list of object classes in the schema installed with Oracle Internet Directory

This section contains these topics:

- Subclasses, Superclasses, and Inheritance
- Object Class Types

Subclasses, Superclasses, and Inheritance

A **subclass** is an object class derived from another object class. The object class from which it is derived is called its **superclass**. For example, the object class `organizationalPerson` is a subclass of the object class `person`. Conversely, the object class `person` is the superclass of the object class `organizationalPerson`.

Subclasses **inherit** all of the attributes belonging to their superclasses. For example, the subclass `organizationalPerson` inherits the attributes of its superclass, `person`. Entries may inherit the attributes defined by multiple object classes.

Note: In itself, an object class contains no values. Only an instance of an object class—that is, an entry—contains values. When a subclass inherits attributes from a superclass, it inherits only the attribute framework—not the attribute values—of the superclass.

One special object class, called `top`, has no superclasses. It is one of the superclasses of every structural object class in the directory, and its attributes are inherited by every entry.

Object Class Types

There are three types of object classes:

- Abstract
- Structural
- Auxiliary

Abstract Object Classes

An abstract object class is a virtual object class. It is used only for convenience when specifying the highest levels of the object class hierarchy. It cannot be the only object class for an entry. For example, the object class `top` is an abstract object class. It is required as a superclass for all structural object classes, but it cannot be used alone.

The `top` object class includes the mandatory attribute `objectClass` as well as several optional attributes. The optional attributes in `top` are:

- `orclGuid`—Global identification which remains constant if the entry is moved
- `creatorsName`—Name of the creator of the object class
- `createTimestamp`—Time when the object class was created
- `modifiersName`—Name of the last person to modify the object class
- `modifyTimestamp`—Time when the object class was last modified
- `orclACI`—**access control list (ACL)** directives that apply to all entries in the subtree below the **access control policy point** where this attribute is defined
- `orclEntryLevelACI`—Access control policy pertaining to only a specific entity—for example, a special user

> **See Also:** "Globalization Support" on page 2-14 for more information on access control policies and ACLs.

Structural Object Classes

Structural object classes describe the basic aspects of an object. Most of the object classes that you use are structural object classes, and every entry should belong to at least one structural object class. Examples of structural object classes are `person` and `groupOfNames`.

These object classes use structure rules to place restrictions on the kinds of object classes you can create under any given object class. For example, a structure rule might require all objects below the `organization` (`o`) object class to be organizational units (ou). Following this rule, you could not enter person objects directly below an organization object class. Similarly, a structure rule might disallow you from placing an organizational unit (ou) object below a person object.

Auxiliary Object Classes

Auxiliary object classes are groupings of attributes that expand the existing list of attributes in an entry. For example, suppose you have defined an entry as a member of two object classes, and you want to assign to that entry additional attributes that do not belong to either of those two object classes. You can create a new auxiliary object class containing the extra attributes, and then associate that auxiliary object class with the entry. This is an alternative to redefining existing object classes.

Unlike structural object classes, auxiliary classes do not place restrictions on where an entry may be stored.

Note: Oracle Internet Directory does not enforce structure rules. It therefore handles both structural and auxiliary object classes in the same way.

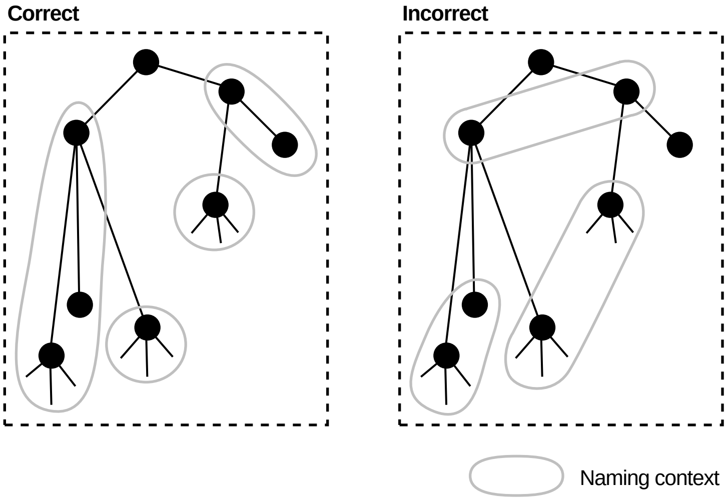
See Also: Chapter 7, "Managing the Directory Schema."

Naming Contexts

A **naming context** is a subtree that resides entirely on one server. It must be contiguous, that is, it must begin at an **entry** that serves as the top of the subtree, and extend downward to either leaf entries or references to subordinate naming contexts. It can range in size from a single entry to the entire **DIT**.

Figure 2–3 illustrates valid and invalid naming contexts. Notice that the correct ones on the left are contiguous, and the incorrect ones on the right are not.

Figure 2–3 Valid and Invalid Naming Contexts



To enable users to search for specific naming contexts, you can publish those naming contexts by using either Oracle Directory Manager or ldapmodify.

> **See Also:** "Managing Naming Contexts" on page 6-16 for instructions on how to publish a naming context

The Directory Schema

The directory **schema** contains all information about how data is organized in the DIT—that is, metadata such as that for an **object class**, an **attribute**, a **matching rule**, and syntax. The directory schema stores this information in a special class of entry called a **subentry**. Oracle Internet Directory, following LDAP Version 3 standards, holds schema definitions in the subentry called `subSchemaSubentry`.

You can add new object classes and objects by modifying `subSchemaSubentry`. You cannot, however, add new matching rules and syntaxes beyond those already supported by Oracle Internet Directory.

See Also:

- Chapter 7, "Managing the Directory Schema."
- Appendix C for a list of both standard and proprietary schema elements installed with Oracle Internet Directory

Security

Oracle Internet Directory provides many powerful features for protecting information. These include:

- Data integrity: Ensuring that data is not modified during transmission
- Data privacy: Ensuring that data is not inappropriately detected during transmission
- Authentication: Ensuring that the identities of users, hosts, and clients are correctly validated
- Authorization: Ensuring that a user reads or updates only the information for which that user has privileges
- Password policies: Establishing and enforcing rules for how passwords are defined and used
- Password protection: Ensuring the security of passwords.

More significantly, in an enterprise or hosted environment, you can use all these features to control access to application metadata—the information governing how applications behave and who can access them. To do this, you deploy the directory for **administrative delegation**. This deployment allows, for example, a global administrator to delegate to department administrators access to the metadata of applications in their departments. These department administrators can then control access to their department applications.

> **See Also:** Chapter 11, "About Security in Oracle Internet Directory" for a fuller discussion of the security features of Oracle Internet Directory

Globalization Support

Oracle Internet Directory follows LDAP Version 3 internationalization (I18N) standards. These standards require that the database storing directory data use the **UTF-8** (Unicode Transformation Format 8-bit) character set. This allows Oracle Internet Directory to store the character data of almost any language supported by Oracle Globalization Support. Moreover, although several different **application program interfaces** (**APIs**) are involved in the Oracle Internet Directory implementation, Oracle Internet Directory ensures that the correct character encoding is used with each API.

Globalization Support uses both single-byte and multibyte characters. A single-byte character is represented by one byte of memory. ASCII text, for example, uses single-byte characters. By contrast, a multibyte character can be represented by more than one byte. Simplified Chinese, for example, uses multibyte characters. A directory entry in simplified Chinese might look like this:

```
dn: o=\274\327\271\307\316\304,c=\303\300\271\372
objectclass: top
objectclass: organization
o: \274\327\271\307\316\304
```

where the attribute values correspond to character strings in the simplified Chinese character set.

The main Oracle Internet Directory components—OID Monitor (OIDMON), OID Control Utility (OIDCTL), Oracle directory server (OIDLDAPD), Oracle directory replication server (OIDREPLD), and the Oracle directory integration server (ODISRV)—always use the UTF-8 character set by default.

Oracle Directory Manager, a Java-based tool, internally uses **Unicode** (**UCS-2**—that is, fixed-width 16-bit Unicode). In Java, UCS-2 is the easiest way to handle characters—including English characters. The Java client uses standard Java packages to convert both to and from UCS-2 and UTF-8. This enables Oracle Directory Manager to handle the LDAP Version 3 protocol using UTF-8.

See Also:

- "Oracle Internet Directory Architecture" on page 2-15 for information on the main Oracle Internet Directory components
- Chapter 9, "Managing Globalization Support in the Directory" for instructions on using Globalization Support in Oracle Internet Directory
- *Oracle9i Globalization and National Language Support Guide* for a detailed discussion of Globalization Support

Oracle Internet Directory Architecture

This section contains these topics:

- An Oracle Internet Directory Node
- An Oracle Directory Server Instance
- Configuration Set Entries

An Oracle Internet Directory Node

Figure 2–4 on page 2-17 shows the various directory server components and their relationships running on a single node.

Oracle Net Services is used for all connections between the Oracle database server and:

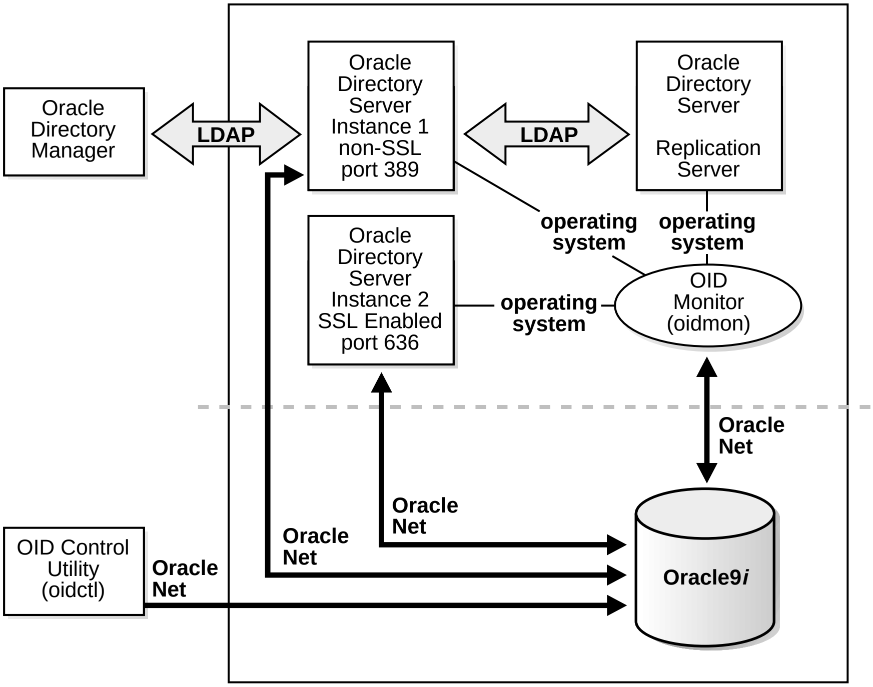
- The **OID Control Utility**
- The Oracle directory server instance 1 non-SSL port 389
- The Oracle directory server instance 2 SSL-enabled port 636
- The **OID Monitor**

LDAP is used for connections between directory server instance 1 on non-SSL port 389 and:

- Oracle Directory Manager
- Oracle directory replication server

The two Oracle directory server instances and the Oracle directory replication server connect to OID Monitor by way of the operating system.

Figure 2–4 A Typical Oracle Internet Directory Node


Oracle Directory Manager
LDAP
Oracle Directory Server Instance 1 non-SSL port 389
LDAP
Oracle Directory Server Replication Server
operating system
operating system
Oracle Directory Server Instance 2 SSL Enabled port 636
operating system
OID Monitor (oidmon)
Oracle Net
Oracle Net
Oracle Net
OID Control Utility (oidctl)
Oracle Net
Oracle9i

Note: In Figure 2–4, the database is on the same node as the directory server processes. However, because all connections with the database are through **Oracle Call Interface (OCI)** and **Oracle Net Services**, it is possible to use a database on a different server.

An Oracle Internet Directory node (Figure 2–4) includes the following major components:

Component	Description
Oracle directory server instance	Also called either an LDAP server instance or a directory server instance. A directory server instance services directory requests through a single Oracle Internet Directory dispatcher process listening at a specific TCP/IP port number. There can be more than one directory server instance on a node, each listening on a different port. One instance comprises one dispatcher process and one or more server processes. By default, there is one server process for each instance, but you can increase this number. Oracle Internet Directory dispatcher and server processes can use multiple threads to distribute the load.
Oracle directory replication server	Also called a replication server. It tracks and sends changes to replication servers in another Oracle Internet Directory system. There can be only one replication server on a node. You can choose whether or not to install and use the replication server.
Oracle9*i* database	Stores the directory data. Oracle Corporation strongly recommends that you dedicate a database for use by the directory. The database can reside on the same node as the servers or on a separate node.

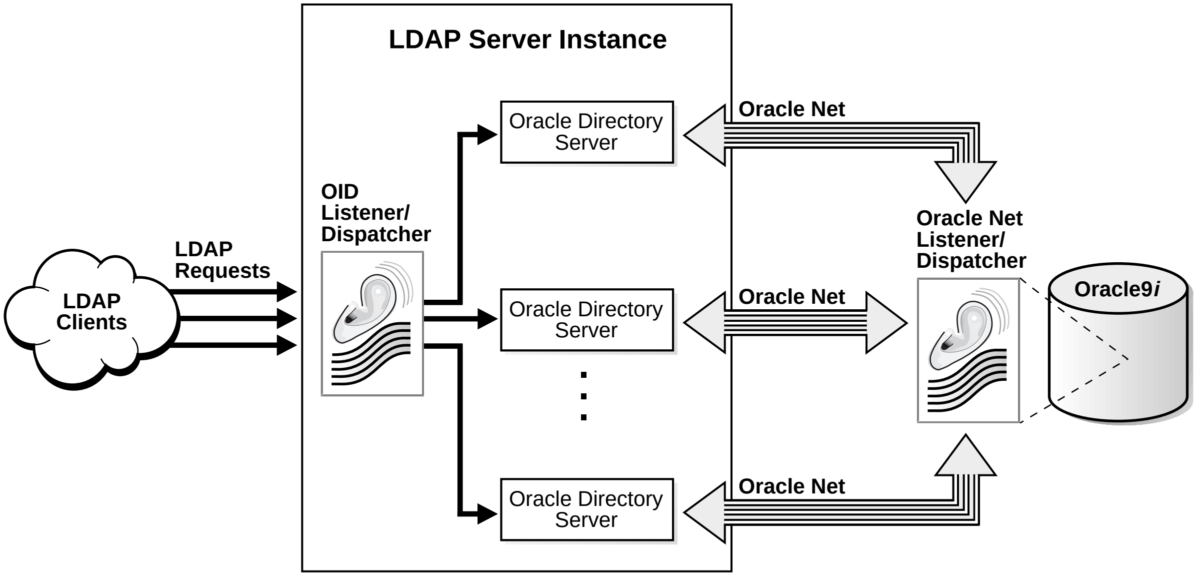
Component	Description
OID Monitor (OIDMON)	Initiates, monitors, and terminates the LDAP server processes. If you elect to install a replication server, OID Monitor controls it. When you issue commands through OID Control Utility (OIDCTL) to start or stop directory server instances, your commands are interpreted by this process. OID Monitor executes the LDAP server instance startup and shutdown requests that you initiate from OID Control Utility. OID Monitor also monitors servers and restarts them if they have stopped running for abnormal reasons. When it starts a server instance, OID Monitor adds an entry into the directory instance registry and updates data in a process table. When it shuts down the directory server instance, it deletes the registry entry as well as the data corresponding to that particular instance from the process table. If OID Monitor restarts a server that has stopped abnormally, it updates the registry entry with the start time of the server. All OID Monitor activity is logged in the file `ORACLE_HOME/ldap/log/oidmon.log`. This file is on the Oracle Internet Directory server file system. OID Monitor checks the state of the servers through mechanisms provided by the operating system.
OID Control Utility (OIDCTL)	Communicates with OID Monitor by placing message data in Oracle Internet Directory server tables. This message data includes configuration parameters required to run each Oracle directory server instance.

The Oracle directory replication server uses LDAP to communicate with an Oracle directory (LDAP) server instance. To communicate with the database, all components use OCI/Oracle Net Services. Oracle Directory Manager and the command-line tools communicate with the Oracle directory servers over LDAP.

An Oracle Directory Server Instance

Each Oracle directory server instance, also called an LDAP server instance, looks similar to what Figure 2–5 illustrates.

Figure 2–5 Oracle Directory Server Instance Architecture



LDAP clients send LDAP requests to an Oracle Internet Directory listener / dispatcher process listening for LDAP commands at its port.

The OID listener / dispatcher sends the request to the Oracle directory server which, in turn creates server processes. Multiple server processes enable Oracle Internet Directory to take advantage of multiple processor systems. The number of server processes created is determined by the configuration parameter `ORCLSERVERPROCS`. The default is 1 (one). A worker thread for each operation processes the client request.

Database connections from each server process are spawned as needed, depending on the value set for the configuration parameter `ORCLMAXCC`. The default value for this parameter is `10`. The server processes communicate with the data server by way of Oracle Net Services. A Oracle Net Services Listener / Dispatcher relays the request to the Oracle9*i* database server.

Configuration Set Entries

The configuration parameters for each Oracle directory server instance are stored in a directory entry called a configuration set entry, or configset. A configuration set entry holds the configuration parameters for a specific instance of the directory server. When you start an instance of a server by using the OID Control Utility, the start-command you enter contains a reference to one of these configsets and uses the information it contains.

The Oracle directory server is installed with a default configuration set entry (`configset0`) so that you can run the directory server immediately. You can create customized configuration set entries by adding new ones that change specific parameters to meet your needs. You can view, add, and modify these entries by using either **Oracle Directory Manager** or the appropriate command-line tool.

See Also:

- "Managing Server Configuration Set Entries" on page 6-2
- "Configuration Set Entry Attributes" on page C-5 for a list of configuration set entry attributes

Example: How Oracle Internet Directory Works

This example shows you how Oracle Internet Directory processes a search request.

1. The user or client enters a search request that is conditioned by one or more of the following options:
   - SSL: The client and server can establish a session that uses SSL encryption and authentication, or SSL encryption only. If SSL is not used, the client’s message is sent in clear text.
   - Type of user: The user can seek access to the directory either as a particular user or as an anonymous user, depending on which of the two has the necessary privileges to perform the desired function.
   - Filters: The user can narrow the search by using one or more search filters, including those that use the Boolean conditions "and," "or," and "not," and those that use other operators such as "greater than, "equal to," and "less than".
2. If the user or client issues the command by using Oracle Directory Manager, then the latter invokes a query function in the Java Native Interface which, in turn, invokes a function in the C API. If the user or client uses a command-line tool, then the tool directly invokes a C function in the C API.

3. The C API, using the LDAP protocol, sends a request to a directory server instance to connect to the directory.
4. The directory server authenticates the user, a process called binding. The directory server also checks the Access Control Lists (ACLs) to verify that the user is authorized to perform the requested search.
5. The directory server converts the search request from LDAP to Oracle Call Interface (OCI) / Oracle Net Services and sends it to the Oracle9*i* database.
6. The Oracle9*i* database retrieves the information and passes it back through the chain—to the directory server, then to the C API, and, finally, to the client.

Distributed Directories

Although an online directory is logically centralized, it can physically distribute its data onto several servers. This reduces the work a single server would otherwise have to do, and enables the directory to accommodate a larger number of entries.

A distributed directory can be either replicated or partitioned. When information is replicated, the same naming contexts are stored by more than one server. When information is partitioned, each directory server stores one or more unique, non-overlapping naming contexts. In a distributed directory, some information may be partitioned and some may be replicated.

This section contains these topics:

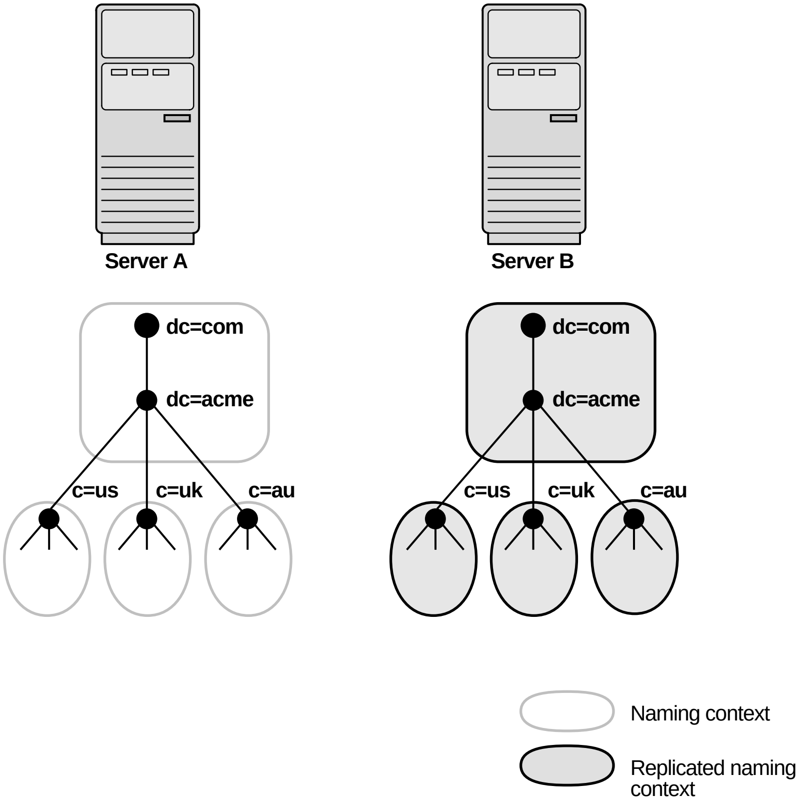
- Replication
- Partitioning

Replication

Replication, in which the same naming contexts are stored by more than one server, improves performance by providing more servers to handle queries. It improves reliability by eliminating risks associated with a single point of failure.

Figure 2–6 shows a replicated directory.

Figure 2–6 A Replicated Directory



Each copy of a naming context that is contained within a server is called a replica. A directory server can hold both read-only and updatable replicas. Servers that hold updatable replicas are called suppliers. Their changes are propagated to other servers called consumers.

At times, the replication process may be unable to apply a change. For example, suppose that Supplier Node A sends the consumer a change, and, immediately after that, Supplier Node B sends it an update to the same entry. Then, suppose that a problem delays the transmission of the entry from Supplier Node A, but that no such problem delays transmission of the update from Supplier Node B. The result can be that the update from Supplier Node B arrives at the consumer ahead of the entry it is modifying. In this case, the replication server makes a specified number of retries to apply the change. If it fails to apply the change once that number is reached, then it moves the change to the human intervention queue, and attempts to apply the change at regular, less frequent intervals that you specify.

Note: This release of Oracle Internet Directory enables replication at the level of the naming context. It does not support replication of part of a naming context.

Also, although there are no Internet standards for directory replication yet, such standards are being developed by the IETF. Oracle Internet Directory replication adheres to the IETF standard proposal for representing directory change information in **change logs**.

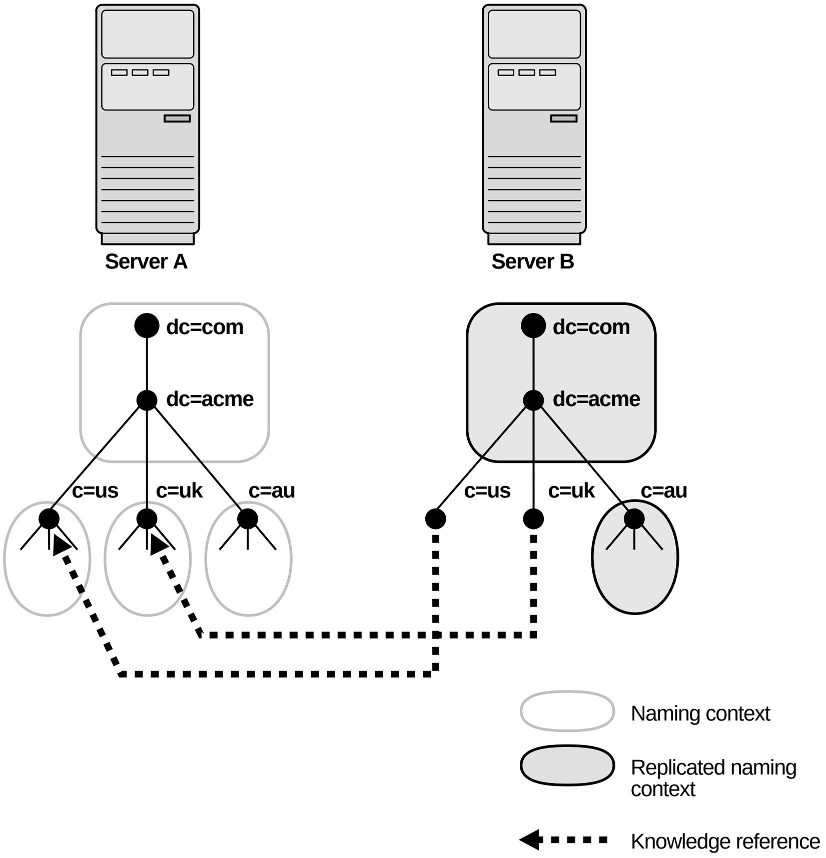
See Also: Chapter 14, "About Directory Replication" for a more detailed discussion of replication, including: Oracle9*i* Replication architecture, change log purging, conflict resolution, and the replication process

Partitioning

Partitioning, in which each directory server stores one or more unique, non-overlapping naming contexts, is another way of distributing directory information.

Figure 2–7 shows a partitioned directory in which some naming contexts reside on different servers.

Figure 2–7 A Partitioned Directory



In Figure 2–7, four naming contexts reside on Server A:

- `dc=acme,dc=com`
- `c=us`
- `c=uk`
- `c=au`

Two naming contexts on Server A are replicated on Server B:

- `dc=acme,dc=com`
- `c=au`

The directory uses **knowledge references**, also called **referrals**, to locate information that is requested of Server B, but that resides on Server A.

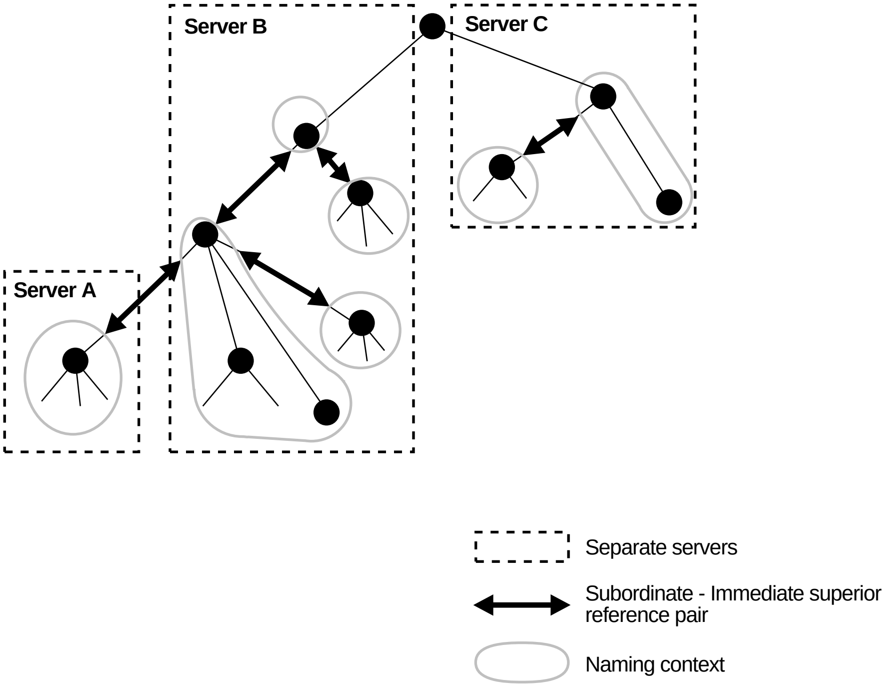
About Knowledge References (Referrals)

Knowledge references provide the names and addresses of the various naming contexts. In Figure 2–7, Server B uses knowledge references to tell clients that Server A has the requested information in the c=us and c=uk naming contexts. Clients can then use the referral information to contact Server A.

Typically, each directory server contains both superior and subordinate knowledge references. Superior knowledge references point upward in the DIT toward the root. They tie the partitioned naming context to its parent. Subordinate knowledge references point downward in the DIT to other partitions.

For example, in Figure 2–8, Server B holds four naming contexts, two of which are superior to the others. These two superior naming contexts use subordinate knowledge references to point to their subordinate naming contexts. Conversely, the naming context on Server A has an immediate superior residing on Server B. Server A therefore uses a superior knowledge reference to point to its parent on Server B.

Figure 2–8 Using Knowledge References to Point to Naming Contexts



Naming contexts that start at the top of the DIT obviously cannot have a knowledge reference to a superior naming context.

Note: There are presently no Internet standards for enforcing the validity of knowledge references, and Oracle Internet Directory does not do so. It is up to the administrator to ensure consistency among knowledge references within an enterprise network.

Oracle Corporation recommends that permission for managing knowledge reference entries be restricted like any other privileged administrative function such as schema or access control.

Kinds of Knowledge References

There are two kinds of knowledge reference:

Smart knowledge reference

Returned when the knowledge reference entry is in the scope of the search. It points the user to the server that stores the requested information.

For example, suppose that:

- Server A holds the naming context `ou=server development,c=us,o=acme`, and has a knowledge reference to Server B
- Server B holds the naming context `ou=sales,c=us,o=acme`

When a user sends a request to Server A for information in `ou=sales,c=us,o=acme`, Server A provides the user with a knowledge reference pointing to Server B.

Default knowledge reference

Returned when the base object is not in the directory, and the operation is performed in a naming context not held locally by the server. A default knowledge reference typically sends the user to a server that has more knowledge about the directory partitioning arrangement.

For example, suppose that Server A holds:

- The naming context `c=us,o=acme`
- A knowledge reference to Server PQR that has more knowledge about the overall directory partitioning arrangement

Now suppose that a client requests information on `c=uk,o=acme`. When Server A finds that it does not have the `c=uk,o=acme` naming context, it points the user to Server PQR. From there, the user can find the server holding the requested naming context.

See Also: "Managing Knowledge References (Referrals)" on page 8-21

The Delegated Administration Service

The Delegated Administration Service enables directory users to modify their own personal data—such as addresses, phone numbers, and photos—without the intervention of an administrator. It also enables users to search other parts of the directory to which they have access, thereby freeing administrators for other tasks in the enterprise.

The Delegated Administration Service relies on an Apache Web server enabled for small Java programs, called servlets, which do the following:

1. Receive requests from clients
2. Process those requests—by either retrieving or updating data in Oracle Internet Directory—then generate results
3. Send responses back to clients

The Oracle Directory Integration Platform

The Oracle Directory Integration platform enables you to synchronize various directories with Oracle Internet Directory. It also makes it easier for third party metadirectory vendors and developers to develop and deploy their own connectivity agents.

This section contains these topics:

- About Metadirectories
- About the Oracle Directory Integration Platform Environment

About Metadirectories

Enterprises today often deploy multiple directories to store information for applications such as ERP systems, database applications, messaging systems, and Network Operating Systems (NOS). Managing so many different directories has many drawbacks, including:

- Increased cost—Multiple administrators must maintain essentially the same information in many different places.
- Inconsistent data—Updated information in one directory is not available to all the other directories.

A metadirectory solves these problems by synchronizing information between all enterprise directories, forming one virtual directory. It centralizes administration, thereby reducing administrative costs, and ensures that data is consistent and up-to-date across the enterprise.

For example, in a metadirectory environment, you can create a global directory entry for each employee. You can populate this entry with data from various synchronized directories—for example, Human Resources applications, messaging systems, or NOS databases. Users can then access this global entry, knowing that the data it contains is up-to-date and synchronized with each **connected directory**.

You can also ensure that the synchronization process respects all existing data ownership policies. For example, you can grant to only the Human Resources department the privilege to change the value of an employee's salary attribute.

About the Oracle Directory Integration Platform Environment

The Oracle Directory Integration platform enables you to:

- Import data from connected directories into Oracle Internet Directory, either all at once or incrementally
- Export data from Oracle Internet Directory into connected directories, either all at once or incrementally
- Synchronize all or part of the data in a connected directory with Oracle Internet Directory. For example, you can decide to synchronize the user name attributes, but not the salary attributes, for enterprise employees.

In an Oracle Directory Integration platform environment, each connected directory synchronizes with Oracle Internet Directory, which serves as the central directory. This provides:

- Consistent, up-to-date information for both users and applications
- A single point of access to all directory data through standards-based clients—for example, Web browsers or email clients
- A central point from which to administer all enterprise directories

See Also: Part VII: "The Oracle Directory Integration Platform"

3

General Deployment Considerations

This chapter discusses issues to consider when deploying Oracle Internet Directory. It helps you assess enterprise directory requirements and make effective deployment choices. Although the recommendations in this chapter are primarily for directories in medium to large enterprises and Internet Service Providers (ISPs), the principles apply to other environments as well.

This chapter contains these topics:

- The Expanding Role of Directories
- Logical Organization Of Directory Information
- Physical Distribution: Partitions and Replicas
- Failover Considerations
- About Capacity Planning, Sizing, and Tuning
- Running Multiple Installations of Oracle Internet Directory on One Host

See Also:

- Chapter 17, "Capacity Planning Considerations" for more detailed information about capacity planning
- Chapter 18, "High Availability And Failover Considerations" for more detailed information about high availability
- Chapter 19, "Tuning Considerations" for more detailed information about tuning
- Part VI: "The Directory and Clusters" for information about failover in clustered environments

The Expanding Role of Directories

Today, most enterprises are at various stages of deploying centralized and consolidated LDAP-compliant directories. Some have had non-LDAP-compliant directories—for example, NDS or ISO X.500—and are now converting to the corresponding LDAP-enabled versions. This is either to accommodate LDAP-reliant Internet clients, such as those embedded in Web browsers, or to consolidate the increasing number of platforms and services that use directories.

The increased numbers of LDAP-enabled applications make availability and performance requirements for LDAP-compliant directories critical. Most environments need to update their deployments.

Enterprises should plan a robust and flexible deployment to accommodate:

- The increased volume of information in the directory
- The number of applications that rely on the directory
- Such load characteristics as concurrent access and throughput

As the directory becomes more central to the operation of the network and its services, deployment choices become critical.

Logical Organization Of Directory Information

Establishing an effective policy for **directory information tree (DIT)** structure and naming requires enterprise-wide coordination and planning. For example, the following questions can arise:

- How do you choose your enterprise directory naming and organization?
- Should the choice reflect the corporate organizational structure or geographic and national boundaries?
- Does the choice work seamlessly for NOS directories such as Novell eDirectory solution and Microsoft Active Directory?

This section contains these topics:

- Directory Entry Naming
- DIT Hierarchy and Structure

Directory Entry Naming

Typically, most enterprises have a Human Resources department that establishes rules for assigning unique names and numbers for employees. When choosing a unique naming component for directory entries, it is good to exploit this administrative infrastructure and use its policies. The alternative, attempting to make DNs more "user friendly," is outweighed by the proliferation of administrative policies it would require.

DIT Hierarchy and Structure

A DIT is hierarchical in structure, similar to the DNS (Domain Name System). It is possible to organize the DIT to reflect any logical hierarchy associated with an enterprise. The choice should accommodate the following:

- The DIT structure and naming policies for the enterprise as a whole should be compatible with the rules and restrictions of departmental NOS directories. For example, some directory products define domains, and then require organizational units and localities to be logically subordinate to those domains. Also, some directory products require directory name uniqueness within a domain, even for entries that are not **siblings**.
- The directory organization should facilitate clear and effective access control and replication policies. In an enterprise where delegation of **ACL** administration is required, it is better to organize the DIT to reflect the data ownership boundaries.

  For example, consider a corporation which has an autonomous data center for each major geographic region: one for the Americas (North and South), one for Europe, and one for Asia Pacific. Suppose that this corporation wants to consolidate its global directory, while retaining the administrative autonomy of its regional data centers. It should organize the directory in **naming contexts** corresponding to each region. This makes it easier to develop access control and replication policies that suit regional needs.
- It may be tempting to organize the directory hierarchy to reflect either the corporate divisional structure or the organizational hierarchy. Usually, this is not advisable because most corporations undergo frequent reorganization and divisional restructuring. It is more manageable to capture a person’s organizational information as an attribute of the person’s directory entry.

Physical Distribution: Partitions and Replicas

You can distribute directory data in two ways:

- By maintaining the entire directory on one server
- By hosting different naming contexts on different servers and connecting one to another by using a **knowledge reference**, also called a referral

See Also: "Distributed Directories" on page 2-22

This section contains these topics:

- An Ideal Deployment
- Partitioning Considerations
- Replication Considerations

An Ideal Deployment

In an ideal world, it would be simpler and more secure to store all naming contexts in a central consolidated directory server. The problem is that this central directory server would then be a single point of failure.

A simple solution might be to implement redundant LDAP servers and their associated databases. However, even redundancy might not provide the needed connectivity, accessibility, and performance that most global organizations need at all their regions and sites. These requirements might, in fact, call for replicas physically located at various regions across the corporate geography.

If Oracle Internet Directory supported only single-master configuration, then logical consolidation of the directory would be difficult. Each region or group would want to store the master replica for the naming context on which that group relies. Because administrators would need to use a different data management procedure for each partition, this could mean a lack of uniformity in the administrative policies among the partitions.

Fortunately, Oracle Internet Directory's multimaster replication makes logical consolidation of the directory easier. It allows "update anywhere" configurations, which makes consolidating the directory more efficient and less costly than maintaining multiple partitions.

Here is a simple and practical recommendation for a robust centralized corporate directory:

- Establish a network of two or more directory nodes, each holding all the naming contexts. Set up these nodes in a multimaster configuration.
- Deploy these individual nodes, 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, then it is better to locate a dedicated directory server for use by the clients in that region.
- Individually configure each regional server for failover and recovery.

Remember: Even if all the naming contexts are consolidated, you can still achieve administrative autonomy for various logical naming contexts. You do this by establishing appropriate access control policies at the root of each naming context.

> **See Also:** "Failover Considerations" on page 3-7 for a discussion of redundancy

Partitioning Considerations

A directory with too many **partitions** generally has more administrative overhead than benefits. This is because each partition requires you to plan backup, recovery, and other data management functions.

Typically, the reasons for maintaining partitions are:

- They correspond to administrative and data ownership boundaries that are better left independent
- The enterprise network has regions that are connected with expensive or low-speed links and many partitions have only local access needs
- The lack of availability of a partition does not have a larger impact
- Maintaining an entire corporate directory in a certain region is too expensive

When you use partitioning, connect one partition to another by using a **knowledge reference**.

Note: LDAP does not support automatic chaining of knowledge references by the LDAP server. The majority of client side LDAP APIs support client-driven knowledge reference chasing. However, there is no guarantee that knowledge references will be supported in all the LDAP tools. The lack of consistent knowledge reference support across all available tools is a factor to consider before deciding to use partitions.

Replication Considerations

LDAP directory replication architecture is based on a loose consistency model: Two replicated nodes in a **replication agreement** are not guaranteed to be consistent in real time. This increases the overall flexibility and availability of the directory network, because a client can modify data without all interconnected nodes being available. Suppose, for example, that one node is unavailable or heavily loaded. With multimaster replication, the operation can be performed on an alternate node, and all interconnected nodes synchronize in due course.

There are many reasons to implement a replicated network, including the following:

- Local accessibility and performance requirements

  Most corporations have operations in many regions in the world, and those operations need a common directory. Suppose that the regions were interconnected with low bandwidth links involving multiple intermediate routers. A client accessing a directory server from outside the region could experience a very high **latency**, and even inadequate **throughput**.

  In such cases, a regional replica—enabled by multimaster replication to receive updates— is essential. Moreover, the replication data transfer can be scheduled for off-peak hours in the underlying **advanced symmetric replication (ASR)**.

- Load balancing

  When directory access exceeds the capacity of an existing server, an additional server must share the load. With Oracle Internet Directory, two such systems can be deployed in a multimaster replication mode. In fact, even when planning the directory deployment to meet a specific estimated load, it can be less costly to maintain two relatively low-end systems than one high-end system. In addition to load balancing, such configurations also contribute to higher system availability.

- Failure tolerance and higher overall system availability

  One of the most important reasons to implement directory replication is to increase overall system availability. When one server is unavailable, the traffic can be routed to other available servers. This can be transparent to clients.

Failover Considerations

Because a directory service has a critical function in an enterprise, deployment should take failure recovery and high availability into consideration. This includes developing backup and recovery strategies for individual nodes.

In addition to multimaster replication, consider the following failover and high-availability options for potential deployment at any Oracle Internet Directory installation:

- Intelligent Client Failover

  All LDAP clients connecting to Oracle Internet Directory can maintain a list of alternate server instances of Oracle Internet Directory to contact if their connection with a given server instance is abruptly broken.

- Intelligent Network Level Failover

  There are several hardware and software solutions that can detect the failure of the system hosting Oracle Internet Directory. These solutions can intelligently reroute future connection requests to an alternate server. Some of these solutions balance the load of incoming connection requests with alternate servers, while also providing the necessary failover capabilities.

Because Oracle Internet Directory is a client of Oracle9*i*, other failover technologies, such as Oracle Real Application Clusters, are also available.

See Also:

- Chapter 18, "High Availability And Failover Considerations" for further details about high-availability and failover options available with Oracle Internet Directory
- Part VI: "The Directory and Clusters" for information about failover in clustered environments

About Capacity Planning, Sizing, and Tuning

When estimating enterprise-wide and regional requirements for directory usage, plan for future needs. Depending on other configuration choices for replication and failover, there could be more than one directory node, each with its own load and capacity requirements. In this case, you must individually size each directory node.

As an enterprise increases its directory usage, more applications rely on Oracle Internet Directory to serve their requests in a timely manner. Ensure that the Oracle Internet Directory installation can live up to the performance and capacity expectations of those applications.

You can influence the capacity and performance of a given Oracle Internet Directory installation in two phases of the deployment process:

- Planning phase

  During this phase, gather the requirements of all directory users and establish a unified performance and capacity requirement. This consists of capacity planning and system sizing.

- Implementation phase

  Once you have the hardware, tune the Oracle Internet Directory software stack for best use of the hardware resources. This improves the performance of Oracle Internet Directory and of the LDAP client applications.

This section contains these topics:

- Capacity Planning
- Sizing Considerations
- Tuning Considerations

Capacity Planning

Capacity planning is the process of determining performance and capacity requirements. You base these on typical models of directory usage in the enterprise.

When trying to estimate the required capacity of an Oracle Internet Directory installation, consider:

- The type of LDAP client applications
- The number of users accessing those applications
- The nature of LDAP operations those applications perform

- The number of entries in the DIT
- The type of operations performed against the Oracle directory server
- The number of concurrent connections to the Oracle directory server
- The peak rate at which operations need to be performed by the Oracle directory server
- The average latency of operations required under peak load conditions

While estimating these details, allow room for future increases in directory usage.

Sizing Considerations

Once you have established the fundamental capacity and performance requirements, translate them into system requirements. This is called system sizing. Some of the details to consider in this phase are:

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

Based on current experience, the following table indicates the approximate level of CPU power required for various deployment scenarios for Oracle Internet Directory:

Usage	Active Connections	Num CPUs	SPECint_rate95 baseline	System
Departmental	0-500	2	60 to 200	Compaq AlphaServer 8400 5/300 (300Mhz x 2)
Organization wide	500-2000	4	200 to 350	IBM RS/6000 J50 (200MHz x 4)
Enterprise wide	2000+	4+	350+	Sun Ultra 450 (296 MHz x 4)

The amount of disk space required for an installation of Oracle Internet Directory is directly proportional to the number of entries stored in the DIT. The following table gives the approximate disk space requirements for variously sized DITs.

Number of Entries in DIT	Disk Requirements
100,000	450MB to 650MB
200,000	850MB to 1.5GB
500,000	2.5GB to 3.5GB
1,000,000	4.5GB to 6.5GB
1,500,000	6.5GB to 10GB
2,000,000	9GB to 13GB

The data in this table makes the following assumptions:

- There are approximately 20 cataloged attributes
- There are approximately 25 attributes for each entry
- The average size of an attribute is approximately 30 bytes

The amount of memory required for Oracle Internet Directory is mostly governed by the amount of database buffer cache that a deployment site desires. Often, the size of the database buffer cache is directly proportional to the number of entries in the DIT. The following table provides estimates of the memory requirements for various DIT sizes:

Directory Type	Number of Entries	Minimum Memory
Small	Less than 600,000	512MB
Medium	600,000 to 2,000,000	1GB
Large	Greater than 2,000,000	2GB

See Also: Chapter 17, "Capacity Planning Considerations."

Tuning Considerations

Oracle Corporation recommends that you properly tune Oracle Internet Directory before using it in a production environment. Before tuning, ensure that there are adequate testing mechanisms and sample data in the directory to simulate a real world usage scenario. Perhaps you can use the applications that rely on the directory for testing purposes.

Any tool for testing the performance of Oracle Internet Directory must be able to show:

- The overall throughput it is noticing
- The average latency of operations

In this way, the tool provides a feedback mechanism for determining the effects of tuning and providing direction to the overall tuning effort.

Some of the commonly tuned properties of an Oracle Internet Directory installation include:

- CPU usage

  This is determined, to a large extent, by:

  - The number of Oracle directory servers
  - The number of database connections opened by each server

  On the one hand, too large a number of Oracle directory servers and database connections can cause too much contention for available CPU resources. On the other hand, too small a number of Oracle directory servers and database connections can leave much of the CPU power under-utilized. Consider adjusting these numbers to the appropriate levels based on available CPU resources and the expected peak load.

- Memory usage

  The main consumer of memory in an Oracle Internet Directory installation is the database buffer cache, which is part of the **SGA**. In some cases, allocating a very large database buffer cache can eliminate much disk I/O for Oracle data files. However, it can also cause paging, which is detrimental to performance. Alternatively, having a small database buffer cache causes too much disk I/O, and that is also detrimental to performance. Tune the memory usage of the system so that all consumers of memory in the system can get physical memory without needing to use paging.

- Disk usage

  Because all of the data served by Oracle Internet Directory resides in database tablespaces, pay attention to any tuning that can increase the I/O throughput. Common techniques for disk tuning include:

  - Balancing tablespaces on different logical and physical drives
  - Striping logical volumes onto multiple physical volumes
  - Distributing disk volumes across multiple I/O controllers

  See Also: Chapter 19, "Tuning Considerations" for further details on various tuning tips and techniques

Running Multiple Installations of Oracle Internet Directory on One Host

You can run more than one installation of Oracle Internet Directory on a single host and then replicate between them. This can be useful in providing up-to-date directory data on the same machine by automatically backing up that data. It also enables you to provide for failover by using only two nodes: If one node fails, then both instances of Oracle Internet Directory can run on the other node.

See Also: "Identifying a Node as Independent of Its Host" on page 15-32 to configure replication between two Oracle Internet Directory installations on the same host

4

Preliminary Tasks

This chapter guides you through some tasks you must perform before configuring and using Oracle Internet Directory, namely, starting the OID Monitor and starting a directory server instance. You also need to reset the default security configuration and reset the password for the database.

This section contains these topics:

- Task 1: Start the OID Monitor
- Task 2: Start a Server Instance
- Task 3: Reset the Default Security Configuration
- Task 4: Reset the Default Password for the Database

Task 1: Start the OID Monitor

The OID Monitor must be running to process commands to start and stop the server.

> **Note:** Although you can start the directory server without using OID Monitor and the OID Control Utility, Oracle Corporation recommends that you use them. This way, if the directory server unexpectedly terminates, then OID Monitor automatically restarts it.

This section contains these topics:

- Starting the OID Monitor
- Stopping the OID Monitor

Starting the OID Monitor

To start the OID Monitor:

1. Set the following environment variables:
   - *ORACLE_HOME*
   - ORACLE_*SID* or a proper TNS CONNECT string
   - NLS_LANG (*APPROPRIATE_LANGUAGE*.UTF8). The default language set at installation is AMERICAN_AMERICA.
2. At the system prompt, type:

```
oidmon [connect=net_service_name] [sleep=seconds] start
```

Argument	Description
connect=*net_service_name*	Specifies the net service name of the database to which you want to connect. This is the network service name set in the `tnsnames.ora` file. This argument is optional.
sleep=*seconds*	Specifies number of seconds after which the OID Monitor should check for new requests from OID Control and for requests to restart any servers that may have stopped. The default sleep time is 10 seconds. This argument is optional.
start	Starts the OID Monitor process

For example:

```
oidmon connect=dbs1 sleep=15 start
```

Stopping the OID Monitor

To stop the OID Monitor daemon, at the system prompt, type:

```
oidmon [connect=net_service_name] stop
```

Argument	Description
connect=*net_service_name*	Specifies net service name of the database to which you want to connect. This is the net service name set in the `tnsnames.ora` file.
stop	Stops the OID Monitor process

For example:

```
oidmon connect=dbsl stop
```

Task 2: Start a Server Instance

Once the OID Monitor is running, start a server instance by using the OID Control Utility.

> **Note:** The value for the instance flag in the OID Control Utility should always be greater than or equal to one.

This section contains these topics:

- Starting an Oracle Directory Server Instance
- Stopping an Oracle Directory Server Instance
- Starting an Oracle Directory Replication Server Instance
- Stopping an Oracle Directory Replication Server Instance
- Restarting Directory Server Instances
- Troubleshooting Directory Server Instance Startup

Starting an Oracle Directory Server Instance

The syntax for starting an Oracle directory server instance is:

oidctl connect=*net_service_name* server=oidldapd instance=*server_instance_number* [configset=*configset_number*] [flags=' -p *port_number* -work *maximum_number_of_worker_threads_per_server* -debug *debug_level* -l *change_logging*' -server *number_of_server_processes*] start

Argument	Description
connect=*net_service_name*	If you already have a `tnsnames.ora` file configured, this is the net service name specified in that file, located in `ORACLE_HOME/network/admin`
server=oidldapd	Type of server to start (valid values are OIDLDAPD and OIDREPLD). This is not case-sensitive.
instance=*server_instance_number*	Instance number of the server to start. Should be a number between 1 and 1000.
configset=*configset_number*	Configset number used to start the server. This defaults to `configset0` if not set. This should be a number between 0 and 1000.
-p *port_number*	Specifies a port number during server instance startup. The default port number is 389.
-work *maximum_number_of_worker_threads_per_server*	Specifies the maximum number of worker threads for this server
-debug *debug_level*	Specifies a debug level during Oracle directory server instance startup
-l *change_logging*	Turns replication change logging on and off. To turn it off, enter `-l false`. To turn it on, do one of the following: ▪ omit the `-l` flag ▪ enter simply `-l` ▪ enter `-l true`
-server *number_of_server_processes*	Specifies the number of server processes to start on this port
start	Starts the server specified in the `server` argument.

For example, to start a directory server instance whose net service name is dbs1, using configset5, at port 12000, with a debug level of 1024, an instance number 3, and in which change logging is turned off, type at the system prompt:

```
oidctl connect=dbs1 server=oidldapd instance=3 configset=5 flags='-p 12000
-debug 1024 -l ' start
```

When starting and stopping an Oracle directory server instance, the server name and instance number are mandatory, as are the commands `start` or `stop`. All other arguments are optional.

All keyword value pairs within the flags arguments must be separated by a single space.

Single quotes are mandatory around the flags.

The configset identifier defaults to zero (`configset0`) if not set.

Note: If you choose to use a port other than the default port (389 for non-secure usage or 636 for secure usage), you must tell the clients which port to use to locate the Oracle Internet Directory. If you use the default ports, clients can connect to the Oracle Internet Directory without referencing a port in their connect requests.

Stopping an Oracle Directory Server Instance

OID Monitor must be running whenever you start or stop directory server instances.

At the system prompt, type:

```
oidctl connect=net_service_name server=OIDLDAPD instance=server_instance_number
stop
```

For example:

```
oidctl connect=dbs1 server=oidldapd instance=3 stop
```

Starting an Oracle Directory Replication Server Instance

The syntax for starting the Oracle directory replication server is:

```
oidctl connect=net_service_name server=oidrepld instance=server_instance_number
[configset=configset_number] flags=' -p directory_server_port_number -d debug_
level -h directory_server_host_name
-m [true | false]-z transaction_size ' start
```

Argument	Description
`connect=`*`net_service_name`*	If you already have a `tnsnames.ora` file configured, then this is the name specified in that file, which is located in ORACLE_ HOME/network/admin
`server=oidrepld`	Type of server to start (valid values are OIDLDAPD and OIDREPLD). This is not case-sensitive.
`instance=`*`server_instance_number`*	Instance number of the server to start. Should be a number between 1 and 1000.
`configset=`*`configset_number`*	Configset number used to start the server. The default is `configset0`. This should be a number between 0 and 1000.
`-p` *`directory_server_port_number`*	Port number that the replication server uses to connect to the directory on TCP port *directory_server_port_number*. If you do not specify this option, the tool connects to the default port (389).
`-d` *`debug_level`*	Specifies a debug level during replication server instance startup
`-h` *`directory_server_host_name`*	Specifies the *directory_server_host_name* to which the replication server connects, rather than to the default host, that is, your local computer. *Directory_server_host_name* can be a computer name or an IP address. (Replication server only)
`-m [`*`true`*`|`*`false`*`]`	Turns conflict resolution on and off. Valid values are `true` and `false`. The default is true. (Replication server only)
`-z` *`transaction_size`*	Specifies the number of changes applied in each replication update cycle. If you do not specify this, the number is determined by the Oracle directory server sizelimit parameter, which has a default setting of 1024. You can configure this latter setting.
start	Starts the server specified in the *server* argument.

For example, to start the replication server with an instance=1, at port 12000, with debugging set to 1024, type at the system prompt:

```
oidctl connect=dbs1 server=oidrepld instance=1 flags='-p 12000 -h eastsun11 -d
1024' start
```

When starting and stopping an Oracle directory replication server, the `-h` flag, which specifies the host name, is mandatory. All other flags are optional.

All keyword value pairs within the flags arguments must be separated by a single space.

Single quotes are mandatory around the flags.

The configset identifier defaults to zero (`configset0`) if not set.

> **Note:** If you choose to use a port other than the default port (389 for non-secure usage or 636 for secure usage), you must tell the clients which port to use to locate the Oracle Internet Directory. If you use the default ports, clients can connect to the Oracle Internet Directory without referencing a port in their connect requests.

Stopping an Oracle Directory Replication Server Instance

OID Monitor must be running whenever you start or stop directory server instances.

At the system prompt, type:

```
oidctl connect=net_service_name server=OIDREPLD instance=server_instance_number
stop
```

For example:

```
oidctl connect=dbs1 server=oidrepld instance=1 stop
```

Restarting Directory Server Instances

If you use OID Monitor and the OID Control utility, then you can both stop and restart the directory server in one command, namely, `restart`. This is useful when you want to refresh the server cache immediately, rather than at the next scheduled time. When the directory server restarts, it maintains the same parameters it had before it stopped. You cannot override these original parameters by entering new ones in the restart command.

To restart a directory server instance, at the system prompt, type:

```
oidctl connect=net_service_name server={oidldapd|oidrepld} instance=server_
instance_number  restart
```

OID Monitor must be running whenever you start, stop, or restart directory server instances.

If you try to contact a server that is down, you receive from the SDK the error message 81—LDAP_SERVER_DOWN.

If you change a configuration set entry that is referenced by an active server instance, you must stop that instance and restart it to effect the changed value in the configuration set entry on that server instance. You can either issue the STOP command followed by the START command, or you can use the RESTART command. RESTART both stops and restarts the server instance.

For example, suppose that Oracle directory server instance1 is started, using configset3, and with the net service name dbs1. Further, suppose that, while instance1 is running, you change one of the attributes in configset3. To enable the change in configset3 to take effect on instance1, you enter the following command:

```
oidctl connect=dbs1 server=oidldapd instance=1 restart
```

If there are more than one instance of the Oracle directory server running on that node using configset3, then you can restart all the instances at once by using the following command syntax:

```
oidctl connect=dbs1 server=oidldapd restart
```

Note that this command restarts all the instances running on the node, whether they are using configset3 or not.

Important Note: During the restart process, clients cannot access the Oracle directory server instance. However, the process takes only a few seconds to execute.

Troubleshooting Directory Server Instance Startup

If the directory server fails to start, you can override all user-specified configuration parameters to start the directory server and then return the configuration sets to a workable state by using the ldapmodify operation.

To start the directory server by using its hard-coded default parameters instead of the configuration parameters stored in the directory, type at the system prompt:

```
oidctl connect=net_service_name flags='-p port_number -f'
```

The `-f` option in the flags starts the server with hard-coded configuration values, overriding any defined configuration sets except for the values in `configset0`.

To see debug log files generated by the OID Control Utility, navigate to `$`*`ORACLE_HOME`*`/ldap/log`.

Task 3: Reset the Default Security Configuration

When you first install Oracle Internet Directory, the default configuration grants to all users read, browse, and search access to all entries in the directory. At the very beginning, you need to establish and implement an access control policy to ensure that each user receives the appropriate authorization. Oracle Corporation specifically recommends that you control access to the subentry `subSchemaSubEntry` and its children because these objects contain information about the directory.

Moreover, when you load directory entries, you are creating a hierarchy of directory entries. You must therefore establish:

- Permissions to load entries into this hierarchy
- Directory access for clients that need read, modify, and write access to the directory entries

See Also:

- Chapter 13, "Managing Directory Access Control" for a detailed explanation of access control options and instructions for setting up security
- Chapter 5, "Using the Administration Tools" for information about the administration tools you use to configure security
- Appendix C, "Schema Elements" for syntax and usage notes for the command-line tools

Task 4: Reset the Default Password for the Database

Oracle Internet Directory uses a password when connecting to an Oracle database. The default for this password when you install Oracle Internet Directory is `ODS`. You can change this password by using the OID Database Password Utility.

See Also: "OID Database Password Utility Syntax" on page A-41 for syntax and usage notes

5

Using the Administration Tools

This chapter introduces the various administration tools of Oracle Internet Directory. It discusses the online administration tool, called Oracle Directory Manager, and tells you how to launch it, navigate through it, and connect to directory servers with it. It also introduces the command-line and bulk tools.

This chapter contains these topics:

- Using Oracle Directory Manager
- Using Command-Line Tools
- Using Bulk Tools
- Using the Catalog Management Tool
- Using the OID Database Password Utility
- Using the Replication Tools
- Using the OID Database Statistics Collection Tool
- Administration Tasks at a Glance

Using Oracle Directory Manager

Oracle Directory Manager is a Java-based tool for administering Oracle Internet Directory. This section describes some of its basic features. More specific instructions are found in sections throughout this book that explain how to perform various tasks.

This section contains these topics:

- Starting Oracle Directory Manager
- Connecting to a Directory Server
- Navigating Oracle Directory Manager
- Connecting to Additional Directory Servers
- Disconnecting from a Directory Server
- Performing Administration Tasks by Using Oracle Directory Manager

Starting Oracle Directory Manager

Before you can launch Oracle Directory Manager, you must have a directory **directory server instance** running.

See Also:

- Chapter 4, "Preliminary Tasks" for instructions on starting a server instance
- "Oracle Internet Directory Architecture" on page 2-15 for a conceptual explanation of directory server instances

To start Oracle Directory Manager, follow the instructions for your operating system:

Operating System	Instructions
Windows NT or Windows 95	From the Start menu, click Programs > *ORACLE_HOME* > Oracle Internet Directory > Oracle Directory Manager
Sun Solaris	If you have not set the path, then navigate to `ORACLE_HOME/bin.` Type at the system prompt: `oidadmin`

The first time you start Oracle Directory Manager, an alert tells you that you must connect to a server. Click OK. The Directory Server Connection dialog box appears.

Connecting to a Directory Server

To connect to a directory server:

1. In the Directory Server Connection dialog box, type the name and port number of an available server.

   The default port is 389. You can change the port if you wish. However, if you have an Oracle directory server running on a port that is not the default, then be sure that any clients that use that server are informed of the correct port.

   Click OK. The Oracle Directory Manager Connect dialog box appears.

2. In each field of the Credentials tab page, type the information specific to this server instance as described in the next table.

Field	Description
User	The first time you log in, do so either as the **super user** or anonymously. If you intend to configure SSL features during this session, login as the super user. If you are logging in as the super user, in the User box, type `cn=orcladmin`. If you are logging in anonymously, leave the User box empty. If you have already set up the user's entry by using LDAP command-line tools, you can enter that user's entry in one of two ways: ■ Browse and select that entry by using the button to the right of the User field ■ Type the **distinguished name (DN)** for that user's entry by using the correct format, for example, `cn=Susie Brown,ou=HR,o=acme,c=us`

Field	Description
Password	If you are logging in as the super user and you specified a password for the super user during installation, in the Password box, type the password you specified. Otherwise, type the default password, namely, `welcome`. After you are logged into Oracle Directory Manager and have connected to a directory server, you should change this password to protect the directory. If you are logging in anonymously, leave the Password box empty. If you want to login as a specific directory user, enter the corresponding password. **See Also:** "Managing Super Users, Guest Users, and Proxy Users" on page 6-22 for instructions on how to change the password
Server	From the Server list, select the host containing the directory server to which you want to connect. If you are already connected to a directory server, and you want to connect to one on a different host: **1.** Click the button to the right of the Server field. The Select Directory Servers dialog box displays a list of available servers. **2.** Select a server. **3.** Click OK. To add a directory server to the list: **1.** In the Select Directory Servers dialog box, click Add. The Directory Server Connection dialog box appears. **2.** In the Server field, type the name of the directory server you want to add. **3.** In the Port field, type the port number for the server you want to add. **4.** Click OK. The added directory appears in the list in the Select Directory Server dialog box. To modify a directory server on the list: **1.** Select the directory server you want to modify. **2.** Click Edit. The Directory Server Connection dialog box appears. **3.** Modify the Server and Port fields, then click OK. The modifications for that server appear in the list in the Select Directory Server dialog box.

Field	Description
Port	The default port (389) appears in this field. If there is more than one directory server instance on the same host, each directory server instance has a different port, and that port number appears in this field when you select the directory server instance. To change this port number: **1.** Click the button to the right of the Server field. **2.** In the Select Directory Server dialog box, select the directory server. **3.** Click Edit. The Directory Server Connection dialog box appears. **4.** In the Directory Server Connection dialog box, in the Port field, enter the new port number, then click Ok.
SSL Enabled	Selecting this check box causes all commands you issue by using Oracle Directory Manager to be sent over Secure Sockets Layer (SSL). You can connect to a directory server either with or without SSL. If you connect by using SSL, then Oracle Directory Manager becomes an SSL client. You can connect in this way if both of the following two conditions are met: ■ The server to which you are connecting uses SSL. If that server does not use SSL, and you select this check box, then authentication will fail. ■ You have already created a wallet containing a certificate and a list of trusted certificates.

See Also:

- Chapter 12, "Managing Secure Sockets Layer (SSL)" for instructions on enabling SSL
- Appendix D, "Using Oracle Wallet Manager" for instructions on creating a wallet
- "Entries" on page 2-2 for instructions on formatting distinguished names
- "Configuring SSL Parameters" on page 12-2 for information about changing ports and their impact on security

3. If you selected the SSL Enabled check box on the Credentials tab, then select the SSL tab.

4. Enter the requested data in the fields as described in the next table.

Field	Description
SSL Location	The client wallet used in two-way authentication. If the client wallet is on the local machine, then type the wallet path and file name by using this syntax: `file: absolute_path_name` If the wallet is on another machine, then link to that location and enter the linked path and file name of the wallet.
SSL Password	The password to open the user's wallet
SSL Authentication	Select the authentication level: ■ No SSL Authentication—Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. If you selected the SSL Enabled check box on the Credentials tab, and choose this option, then only SSL encryption / decryption will be used. ■ SSL Client and Server Authentication—Two-way authentication. Both client and server send certificates to each other. ■ SSL Server Authentication—One-way authentication. Only the directory server authenticates itself to the client by sending its certificate to the client.

5. Click Login. Oracle Directory Manager appears.

Navigating Oracle Directory Manager

This section provides an overview of Oracle Directory Manager, and explains the items in the menu bar and the buttons on the toolbar.

Overview of Oracle Directory Manager

Like the directory itself, the navigator pane (left side of the double window interface) has a tree-like structure. When Oracle Directory Manager first opens, the navigator pane shows only one tree item, Oracle Internet Directory Servers. By clicking the plus sign(+) next to the tree item, subcomponents of that tree item appear.

In the right pane, some windows contain buttons labeled Apply and OK. If you press Apply, the changes you have made are committed, and the window remains available for more changes. If you press OK, the changes you have made are committed, and the window closes.

Similarly, some windows have buttons that are labeled Revert and Cancel. If you press Revert, then the changes you have made in that window do not take effect, the original values reappear in the fields, and the window stays open for further work. If you press Cancel, the changes you have made in that window do not take effect, and the window closes.

The Oracle Directory Manager Menu Bar

The next table lists and describes the menus you can access by using the menu bar. Menu items become enabled or disabled depending on the pane or tab page you are displaying.

Menu	Menu Items
File	Create—Adds an object Create Like—Adds a new object by using the object selected in the navigator pane as a template Connect—Connects to a directory server selected in the navigator pane Disconnect—Disconnects from a directory server selected in the navigator pane Exit—Exits Oracle Directory Manager
Edit	Edit—Modifies an object Remove—Removes a selected object Find Object Classes—Searches for an object class

Menu	Menu Items
View	Refresh—Updates data stored in memory to reflect changes in the database Tear-Off—Generates a secondary dialog containing the fields and values displayed in Oracle Directory Manager's right pane. This is useful when comparing two pieces of information.
Operations	Create Object Class—Displays the New Object Class dialog box that you use to add a new object class Create Attribute—Displays the New Attribute Type dialog box that you use to add a new attribute to an entry Create Access Ctrl Point—Displays the New Access Control Point dialog box that you use to add a new **access control policy point**. Create Entry—Displays the New Entry dialog box that you use to add a new directory entry Refresh Entry—Updates data for entries stored in memory to reflect changes in the database Refresh Subtree Entries—Updates the children of entries stored in memory to reflect changes in the database Drop Index—Removes an index from an attribute. When you select this item, an alert asks you to confirm that you want to drop the index. Search ACPs—Enables you to configure ACP searches User Preferences—Displays a dialog box that enables you to: ■ Configure the display of entry search results ■ Establish whether ACPs are displayed whenever Oracle Directory Manager runs, or only as the result of a search
Help	Contents—Displays the Contents tab page of the Help navigator Search for Help On...—Displays the Help Search dialog box that you use to search for words in the online help guide About Oracle Internet Directory—Displays Oracle Internet Directory version information

The Oracle Directory Manager Toolbar

Figure 5–1 and Table 5–1 together illustrate and describe the Oracle Internet Directory toolbar, starting at the left. Buttons become enabled or disabled depending on the pane or tab page you are displaying in Oracle Directory Manager.

Figure 5–1 Oracle Directory Manager Toolbar

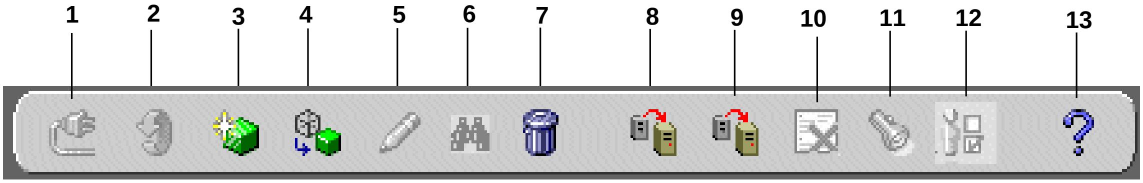


Table 5–1 Oracle Directory Manager Toolbar

Button	Purpose
1	Connect / Disconnect—Connects to or disconnect from a directory server selected in the navigator pane
2	Refresh—Updates data for objects other than entries that are stored in memory to reflect changes in the database
3	Create—Adds a new object
4	Create Like—Adds a new object by using another object as a template
5	Edit—Modifies an object
6	Find Object Classes or Attributes—Searches for either an object class or an attribute, depending on the context. If, in the navigator pane, you navigate to Oracle Internet Directory > *directory_server_instance* > Server Management > Object Classes, then this button searches for an object class. If you navigate to Oracle Internet Directory > *directory_server_instance* > Server Management > Attributes, this button searches for attributes.
7	Delete—Removes an object
8	Refresh Entry—Updates data for entries stored in memory to reflect changes in the database
9	Refresh SubTree Entries—Updates the children of entries stored in memory to reflect changes in the database
10	Drop Index—Removes an index from an attribute. When you click this button, an alert asks you to confirm that you want to drop the index.
11	Search—Enables you to configure ACP searches

Table 5–1 Oracle Directory Manager Toolbar

Button	Purpose
12	User Preferences—Enables you to configure the display of ACPs in the navigator pane, as well as entries in a search operation
13	Help—Displays the Help system

Connecting to Additional Directory Servers

You can connect to more than one directory server at a time, and then view and modify the data, schema, and security for each directory server. If you do this, then each server is listed in the navigator pane under Oracle Internet Directory Servers.

To connect to an additional directory server:

1. In the navigator pane, select Oracle Internet Directory Servers.
2. In the right pane, click New.
3. Follow the login procedures described in "Connecting to a Directory Server" on page 5-3.

Disconnecting from a Directory Server

To disconnect from a directory server by using Oracle Directory Manager, choose File > Disconnect. Also, when you exit Oracle Directory Manager, connections between all directory servers and the directory are automatically disconnected.

All connection information is stored in the user’s home directory in the file `osdadmin.ini`.

When you restart Oracle Directory Manager, all previously connected server connections appear in the Directory Server Login dialog box.

Performing Administration Tasks by Using Oracle Directory Manager

You can perform most of the Oracle Internet Directory administrative tasks through Oracle Directory Manager. Tasks that you cannot perform through Oracle Directory Manager involve running processes, such as starting and stopping the OID Monitor (oidmon) process and starting and stopping server instances. To perform tasks that you cannot perform with Oracle Directory Manager, use the appropriate LDAP command-line tool.

The following table lists the task areas managed by Oracle Directory Manager and where to find instructions for using it in each area.

Task Area	Instructions
Schema administration	"Managing Object Classes by Using Oracle Directory Manager" on page 7-6 "Managing Attributes by Using Oracle Directory Manager" on page 7-17
Entries management	"Managing Entries by Using Oracle Directory Manager" on page 8-2
ACP administration	"Managing Access Control by Using Oracle Directory Manager" on page 13-15
Partitioning and replication	Chapter 15, "Managing Directory Replication"

Using Command-Line Tools

Oracle Internet Directory provides several command-line tools for manipulating directory entries and attributes. This section explains the kind of tasks you can perform with each tool.

Most of the command-line tools act on objects that are in text files written in the LDAP Data Interchange Format (LDIF).

See Also: "LDAP Data Interchange Format (LDIF) Syntax" on page A-2 for information on formatting an LDIF file

The following table lists each command-line tool, the task(s) you can perform with it, and where to find syntax and usage notes.

Tool	Task(s)	Syntax and Usage Notes
ldapadd	Add entries one at a time.	"ldapadd Syntax" on page A-4
ldapaddmt	Add several entries concurrently by using this multithreaded tool.	"ldapaddmt Syntax" on page A-6
ldapbind	Authenticate user/client to a directory server.	"ldapbind Syntax" on page A-8
ldapcompare	See whether an entry contains a specified attribute value.	"ldapcompare Syntax" on page A-9
ldapdelete	Delete entries.	"ldapdelete Syntax" on page A-11
ldapmoddn	Modify the DN or RDN of an entry, rename an entry or a subtree, or move an entry or a subtree under a new parent.	"ldapmoddn Syntax" on page A-13
ldapmodify	Create, update, and delete attribute data for an entry.	"ldapmodify Syntax" on page A-15
ldapmodifymt	Modify several entries concurrently by using this multithreaded tool.	"ldapmodifymt Syntax" on page A-20
ldapsearch	Search for directory entries.	"ldapsearch Syntax" on page A-22

See Also: "Using Globalization Support with Command-Line Tools" on page 9-5 for a discussion of command-line tools and Globalization Support

Using Bulk Tools

Bulk tools enable you to create and manage large numbers of directory entries from data residing in, or created by, other applications.

Important Note: To use these tools you must provide the Oracle Internet Directory password. The default password is ods, although the system administrator can change it by using the OID Database Password Utility.

See Also:

- "Using the OID Database Password Utility" on page 5-14
- "OID Database Password Utility Syntax" on page A-41

The table that follows lists each bulk tool, the task(s) you can perform with it, and where to find syntax and usage notes.

Tool	Task(s)	Syntax and Usage Notes
bulkload	Load large number of entries to Oracle Internet Directory through LDIF files	"bulkload Syntax" on page A-28
ldifwrite	Copy data from the directory information base into an LDIF file that can be read by any LDAP compliant directory server. You can use ldifwrite in conjunction with bulkload. You can also use ldifwrite to back up information from all or part of a directory.	"ldifwrite Syntax" on page A-31
bulkmodify	Modify a large number of existing entries efficiently	"bulkmodify Syntax" on page A-29
bulkdelete	Delete a subtree efficiently	"bulkdelete Syntax" on page A-27

Using OID Control Utility

OID Control Utility is a command-line tool for starting and stopping the server. The commands are interpreted and executed by the OID Monitor process.

See Also:

- "OID Control Utility Syntax" on page A-35
- "Oracle Internet Directory Architecture" on page 2-15 for a conceptual description

Using the Catalog Management Tool

Oracle Internet Directory uses indexes to make attributes available for searches. When Oracle Internet Directory is installed, the entry `cn=catalogs` lists available attributes that can be used in a search. Only those attributes that have an equality matching rule can be indexed.

If you want to use additional attributes in search filters, you must add them to the catalog entry. You can do this at the time you create the attribute by using Oracle Directory Manager. However, if the attribute already exists, then you can index it only by using the Catalog Management tool.

See Also:

- "Catalog Management Tool Syntax" on page A-32 for syntax and usage notes
- "Indexing an Attribute by Using Command-Line Tools" on page 7-30
- "Indexing an Attribute by Using Oracle Directory Manager" on page 7-28

Using the OID Database Password Utility

Oracle Internet Directory uses a password when connecting to an Oracle database. The default for this password when you install Oracle Internet Directory is `ODS`. You can change this password by using the OID Database Password Utility.

See Also: "OID Database Password Utility Syntax" on page A-41 for syntax and usage notes

Using the Replication Tools

When a replication conflict arises, Oracle directory replication server places the change in the retry queue and tries to apply it from there for a specified number of times. If it fails after that specified number, then the replication server puts the change in the human intervention queue. From there, the replication server repeats the change application process at less frequent intervals while awaiting your action.

At this point, you need to:

1. Examine the change in the human intervention queues
2. Reconcile the conflicting changes
3. Place the change either back into the retry queue or into the purge queue.

Two tools assist in this process. Use the OID Reconciliation tool to synchronize conflicting changes, and the Human Intervention Queue Manipulation tool to move changes from the human intervention queue to either the retry queue or the purge queue.

See Also:

- "Using the OID Reconciliation Tool" on page 15-32
- "OID Reconciliation Tool Syntax" on page A-44 for syntax and an explanation of how OID Reconciliation Tool works
- "Using the Human Intervention Queue Manipulation Tool" on page 15-31
- "Human Intervention Queue Manipulation Tool Syntax" on page A-41

Using the OID Database Statistics Collection Tool

The OID database statistics collection tool (oidstats.sh), located in `$ORACLE_HOME/ldap/admin/`, assists in capacity planning. It helps you analyze the various database `ods` schema objects so that you can estimate the statistics.

See Also: "OID Database Statistics Collection Tool Syntax" on page A-47

Administration Tasks at a Glance

Oracle Internet Directory administration tasks are described throughout this manual. The following table points you to the information you need for some of the more common tasks.

Task	Information
Managing Attributes	
Add, modify, or delete an attribute by using command-line tools	"Managing Attributes by Using Command-Line Tools" on page 7-29
Add, modify, or delete an attribute by using the Oracle Directory Manager	"Managing Attributes by Using Oracle Directory Manager" on page 7-17
Managing Entries	
Add, modify, or delete a directory entry by using command-line tools	"Managing Entries by Using Command-Line Tools" on page 8-15
Add, modify, or delete a directory entry by using Oracle Directory Manager	"Managing Entries by Using Oracle Directory Manager" on page 8-2
Import bulk data files	"bulkload Syntax" on page A-28
	"LDAP Data Interchange Format (LDIF) Syntax" on page A-2
View Directory Information Tree (DIT) hierarchy of entries	"Managing Entries by Using Oracle Directory Manager" on page 8-2
Managing Object Classes	
Add, modify, or delete object classes by using command-line tools	"Managing Object Classes by Using Command-Line Tools" on page 7-14
Add, modify, or delete object classes by using Oracle Directory Manager	"Managing Object Classes by Using Oracle Directory Manager" on page 7-6
Managing Replication	
Set up replication	Chapter 15, "Managing Directory Replication"
Resolve replication change conflicts	"Resolving Conflicts Manually" on page 15-30
Move replication changes from human intervention queue to either the retry queue or the purge queue	"Using the Human Intervention Queue Manipulation Tool" on page 15-31
Managing Security	
Set up an Access Control Policy Point (ACP)	Chapter 13, "Managing Directory Access Control"
Set up SSL	Chapter 12, "Managing Secure Sockets Layer (SSL)"

Task	Information
Managing Servers	
Configure server instance parameters by using command-line tools	"Managing Server Configuration Set Entries by Using Command-Line Tools" on page 6-10
Configure server instance parameters by using the Oracle Directory Manager	"Managing Server Configuration Set Entries by Using Oracle Directory Manager" on page 6-4
Connect to a directory by using Oracle Directory Manager	"Connecting to a Directory Server" on page 5-3 "Connecting to Additional Directory Servers" on page 5-10
Start the directory server processes	Chapter 4, "Preliminary Tasks"
Stop the directory server processes	Chapter 4, "Preliminary Tasks"
View system operational attributes	"Setting System Operational Attributes by Using Oracle Directory Manager" on page 6-13 "Setting System Operational Attributes by Using ldapmodify" on page 6-15

Part II

Basic Directory Administration

This part guides you through the tasks to configure and maintain Oracle Internet Directory. This part contains these chapters:

6

Managing the Oracle Directory Server

This chapter explains how to manage an Oracle directory server by using Oracle Directory Manager and command-line tools.

This chapter contains these topics:

- Managing Server Configuration Set Entries
- Setting System Operational Attributes
- Managing Naming Contexts
- Managing Passwords
- Configuring Searches
- Managing Super Users, Guest Users, and Proxy Users
- Setting Debug Logging Levels
- Using Audit Log
- Viewing Active Server Instance Information
- Changing the Password to an Oracle Database Server

See Also: Chapter 4, "Preliminary Tasks" for instructions on starting and stopping directory server instances

Managing Server Configuration Set Entries

When you start an Oracle directory server by using the **OID Control Utility**, that start message refers to a **configuration set entry** containing server parameters. You can add, modify, and delete configuration set entries by using either Oracle Directory Manager or the appropriate command-line tool.

See Also:

- "Configuration Set Entries" on page 2-21 for a conceptual overview of configuration set entries
- "Task 2: Start a Server Instance" on page 4-3 for instructions on how to start the server by using OID Control Utility

This section contains these topics:

- Preliminary Considerations for Managing Configuration Set Entries
- Managing Server Configuration Set Entries by Using Oracle Directory Manager
- Managing Server Configuration Set Entries by Using Command-Line Tools

Preliminary Considerations for Managing Configuration Set Entries

Although you can change values in the default configuration set, namely, `configset0`, all of your changes will be carried over to every new configuration set entry that you create. This is because `configset0` values are used as the template for all new configuration set entries.

When you want to change values that should not always be in effect for every instance of the server that you run, it is better to create new configuration set entries. Note that, in release 3.0.1, this applies to the Oracle directory server instances only. The Oracle replication directory server supports only one configuration set in this release.

You may want to establish a separate instance of a directory server with different values. If you do not want those values to be exercised by all users, set up a new configuration set entry and run a separate server instance pointing to that configuration set entry for groups with special needs.

Figure 6–1 shows three separate directory server instances, each with a different value.

Figure 6–1 Directory Entry Hierarchy Showing Multiple Configuration Set Entries

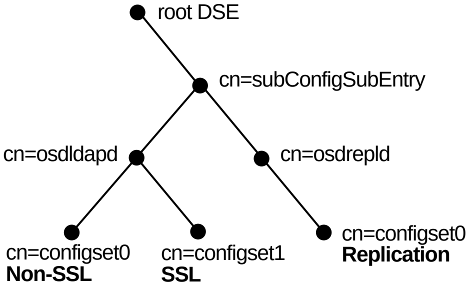


Figure 6–1 shows:

- An Oracle directory server (`cn=osdldap`) with:
  - One instance listening on the default port and using `configset0` with SSL set to *off*
  - A second instance listening on the SSL port and using `configset1` with SSL set to *on*
- A replication server instance (`cn=osdrepld`) using `configset0`

See Also:

- Chapter 12, "Managing Secure Sockets Layer (SSL)" for information about configuration parameters for SSL
- Chapter 15, "Managing Directory Replication" for information about configuration parameters for replication
- "Configuration Set Entry Attributes" on page C-5 for a list and descriptions of the entire set of attributes that are used to configure an instance of a directory server

Managing Server Configuration Set Entries by Using Oracle Directory Manager

You can use Oracle Directory Manager to view, add, modify, and delete configuration set entries.

Important Note: You cannot change the parameters for an active instance directly; you must change the parameters in a configuration set entry and save it. After the configuration set entry is saved, use the OID Control Utility restart command to stop current Oracle directory server instances and restart them.

You can change a configuration set entry and start fresh instances that use the new parameters. The changes will not affect the older instances that are still running, however, unless they have been restarted.

For information on restarting directory server instances, see "Task 3: Reset the Default Security Configuration" on page 4-9.

Viewing Configuration Set Entries by Using Oracle Directory Manager

To view configuration set entries:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server instance* > Server Management, then select Directory Server or Replication Server. The parameters of the active instance appear in the right pane.
2. Choose a specific instance in the right pane. A Server Process dialog box appears.

   You can see all the parameters for the instance by selecting the tabs across the top of the dialog box. However, you cannot change them in this dialog box. To change them, you must change the configuration set entry on which they are based.

   See Also: "Modifying Configuration Set Entries by Using Oracle Directory Manager" on page 6-8

Adding Configuration Set Entries by Using Oracle Directory Manager

The first time you add a configuration set entry, you can:

- Use the default configuration set as a template, then copy from the ones you create to make subsequent configuration sets
- Add a configuration set entry without copying from an existing one

Adding a Configuration Set Entry by Copying from the Default Configuration Set Entry To add configuration set entries by copying the default configuration set entry:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server instance* > Server Management > Directory Server, then select Default Configuration Set.
2. On the toolbar, click the Create Like button. The Configuration Sets dialog box displays the General tab.
3. Fill in the fields with the information described in the following table:

Field	Description
Max. Number of DB Connections	Type the number of concurrent database connections a single directory server process can have. The default is ten.
Number of Child Processes	Type the number of server processes a single instance can spawn. The default is one.
Set	Type the number of the configuration set entry. The default configuration set is 0. There can be as many different configuration sets as needed. The same configuration set can be used by more than one instance if the parameter needs of the multiple instances are the same. The set number is not modifiable.

4. Select the SSL Settings tab and fill in the fields with the information described in this table:

Field	Description
SSL Enable	Select to enable SSL authentication. If you do not select this check box, SSL is not enabled, and you do not need to set any other parameters on this page.

Field	Description
SSL Authentication	Choose one of the following: ■ No SSL Authentication—Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, SSL encryption / decryption only is used. ■ SSL Client and Server Authentication—Both client and server authenticate themselves to each other and send certificates to each other. ■ SSL Server Authentication—Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.

SSL Wallet URL	Type the location of the server-side SSL wallet. If you elect to change the location of the wallet, you must change this parameter. You must set the wallet location on both the client and the server. For example, on Solaris, you could set this parameter as follows:
	`file:/home/my_dir/my_wallet`
	On Windows NT, you could set this parameter as follows:
	`file:C:\my_dir\my_wallet`

SSL Wallet Password	Type the password for the server-side wallet. This password was set during creation of the wallet. If you change the password, you must change this parameter.
SSL Wallet Confirm Password	Retype the new password in this field when you change the password.
SSL Port	The default SSL port is 636. You can change the SSL port.

5. Click Apply.

Note: Remember: The changes will not affect the active directory server instance until you restart it. See "Restarting Directory Server Instances" on page 4-7.

See Also:

- Appendix D, "Using Oracle Wallet Manager" for information about setting the location of the Oracle Wallet and the Oracle Wallet password
- "Setting Debug Logging Levels by Using the OID Control Utility" on page 6-27

Adding a Configuration Set Entry Without Copying from an Existing One To create a new configuration set entry without copying from a previous configuration set entry:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server instance* > Server Management > Directory Server, then select Default Configuration Set.
2. On the toolbar, click Create. A Configuration Sets dialog box displays the General tab page. Fill in the fields as described in this table:

Field	Description
Max. Number of DB Connections	Type the number of concurrent database connections a single directory server process can have. The default is ten.
Number of Child Processes	Type the number of server processes a single instance can spawn. The default is one.
Set	Type the number of the configuration set entry. The default configuration set is 0. There can be as many different configuration sets as needed. The same configuration set can be used by more than one instance if the parameter needs of the multiple instances are the same. The set number is not modifiable.

3. Select the SSL Settings tab and fill in the fields with the information described in this table:

Field	Description
SSL Enable	Select to enable SSL authentication. If you do not select this check box, SSL is not enabled, and you do not need to set any other parameters on this page.

Field	Description
SSL Authentication	Choose one of the following: - No SSL Authentication—Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, SSL encryption / decryption only is used. - SSL Client and Server Authentication—Both client and server authenticate themselves to each other and send certificates to each other. - SSL Server Authentication—Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.

SSL Wallet URL	Type the location of the server-side SSL wallet. If you elect to change the location of the wallet, you must change this parameter. You must set the wallet location on both the client and the server. For example, on Solaris, you could set this parameter as follows:
	`file:/home/my_dir/my_wallet`
	On Windows NT, you could set this parameter as follows:
	`file:C:\my_dir\my_wallet`

SSL Wallet Password	Type the password for the server-side wallet. This password was set during creation of the wallet. If you change the password, you must change this parameter.
SSL Wallet Confirm Password	Retype the new password in this field when you change the password.
SSL Port	The default SSL port is 636. You can change the SSL port.

4. Click Ok.

Modifying Configuration Set Entries by Using Oracle Directory Manager

To modify configuration set entries:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server instance* > Server Management > Directory Server, then select the configuration set entry you want to modify. The configuration set appears in the group of tab pages in the right pane.

   Modify the values in the fields for the General tab as described in this table:

Field	Description
Max. Number of DB Connections	Type the number of concurrent database connections a single directory server process can have. The default is ten.
Number of Child Processes	Type the number of server processes a single instance can spawn. The default is one.
Set	Type the number of the configuration set entry. The default configuration set is 0. There can be as many different configuration sets as needed. The same configuration set can be used by more than one instance if the parameter needs of the multiple instances are the same. The set number is not modifiable.

You can change any of the values. Press Apply to save the changes.

2. Select the SSL Settings tab. Modify the fields as described in the following table.

Field	Description
SSL Enable	Select to enable SSL authentication. If you do not select this check box, SSL is not enabled, and you do not need to set any other parameters on this page.
SSL Authentication	Choose one of the following: ■ No SSL Authentication—Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, SSL encryption / decryption only is used. ■ SSL Client and Server Authentication—Both client and server authenticate themselves to each other and send certificates to each other. ■ SSL Server Authentication—Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.

SSL Wallet URL	Type the location of the server-side SSL wallet. If you elect to change the location of the wallet, you must change this parameter. You must set the wallet location on both the client and the server. For example, on Solaris, you could set this parameter as follows:
	`file:/home/my_dir/my_wallet`
	On Windows NT, you could set this parameter as follows:
	`file:C:\my_dir\my_wallet`

Field	Description
SSL Wallet Password	Type the password for the server-side wallet. This password was set during creation of the wallet. If you change the password, you must change this parameter.
SSL Wallet Confirm Password	Retype the new password in this field when you change the password.
SSL Port	The default SSL port is 636. You can change the SSL port.

3. Once you are satisfied with the parameters you have set for the new configuration set entry, click Apply.
4. Restart the server instance for the command to take effect.

Note: Remember: The changes will not affect the active directory server instance until you restart it. See "Restarting Directory Server Instances" on page 4-7.

See Also: Appendix D, "Using Oracle Wallet Manager" for information on setting the location of the Oracle Wallet and the Oracle Wallet password.

Deleting Configuration Set Entries by Using Oracle Directory Manager

To delete configuration set entries:

1. In the navigator pane, expand Server Management > Directory Server.
2. In the navigator pane, select the configuration set entry you want to delete.
3. Click Delete on the toolbar.

Note: Remember: The changes will not affect the active directory server instance until you restart it. See "Restarting Directory Server Instances" on page 4-7.

Managing Server Configuration Set Entries by Using Command-Line Tools

Although changing configuration set entries by using Oracle Directory Manager is desirable, it can sometimes be more convenient to use the available command-line tools—for example, when you want to make the same set of changes across multiple Oracle directory servers.

When you add or modify configuration set entries by using the command-line tools, the input file for adding a new configuration set entry should be written in **LDAP Data Interchange Format (LDIF)**. It should contain only the attributes and values that differ from the installed defaults. The directory server uses the attribute values that you establish in the new configuration set entry to override its own existing values for these attributes.

> **See Also:** "LDAP Data Interchange Format (LDIF) Syntax" on page A-2 for information on LDIF

Adding Configuration Set Entries by Using ldapadd

If you are adding a new Oracle directory server instance, you can either use an existing configuration set entry, or add a new one for the new instance.

To add a new configuration set entry, create an input file, and then load the input file with ldapadd. Follow these steps:

1. Create the input file in a text editor.

   Input files must use LDIF format. When you create the input file, you need to define or include only those attributes that differ from the current values in that configuration set entry.

   In this example, the parameter `configset2` is the RDN, or local name, of the new entry, the wallet location is: `/HOME/test/wallet`, and the password is `welcome`.

   ```
   dn:cn=configset2, cn=oidldapd, cn=subconfigsubentry
   cn:configset2
   objectclass:orclConfigSet
   objectclass:orclLDAPSubConfig
   objectclass:top
   orclsslauthentication:1
   orclsslenable:1
   orclsslport:5000
   orclsslversion:3
   orclsslwalletpasswd:welcome
   orclsslwalleturl:file:/HOME/test/wallet
   ```

2. Run ldapadd with an input file.

   At the system prompt, type the command to add the input file. If the example shown above were given the file name `newconfigs`, the ldapadd command would look something like this:

```
ldapadd [options] -f newconfigs
```

See Also:

- "LDAP Data Interchange Format (LDIF) Syntax" on page A-2
- "ldapadd Syntax" on page A-4 for a detailed list of options available with this command
- "Configuration Set Entry Attributes" on page C-5 for a description of configuration set entry attributes

Modifying and Deleting Configuration Set Entries by Using ldapmodify

To modify or delete an existing configuration set entry, create an input file containing only the attributes that you want to change, and then load the input file with the ldapmodify command. Follow these steps:

1. Create the input file.

   When you create the input file, define or include only those attributes that differ from the installed defaults.

   Input files must have LDIF format.

   In the example shown below, the parameter `cn=configset2,cn=osdldapd,cn=subconfigsubentry` is the DN, or local name, of an existing configuration set entry. This example shows how to modify the ORCLSSLPORT parameter to 7000.

   ```
   dn:cn=configset2,cn=osdldapd,cn=subconfigsubentry
   changetype: modify
   replace: orclsslport
   orclsslport: 7000
   ```

2. Run ldapmodify referencing the input file.

   Type the command to reference the input file at the system prompt. For example, if the input file were named `configfile`, your ldapmodify command would look something like the command shown that follows:

   ```
   ldapmodify [options] -f configfile
   ```

See Also:

- "LDAP Data Interchange Format (LDIF) Syntax" on page A-2
- "ldapmodify Syntax" on page A-15 for a more detailed discussion of ldapmodify, and a list of its options
- "Configuration Set Entry Attributes" on page C-5 for a description of configuration set entry attributes

Setting System Operational Attributes

Operational **attributes**—as opposed to application attributes—pertain to the operation of the directory itself. Some operational information is specified by the directory to control the server—for example, the time stamp for an entry. Other operational information, such as access information, is defined by administrators and is used by the directory program in its processing. You must have superuser privileges to set system operational attributes.

This section contains these topics:

- Setting System Operational Attributes by Using Oracle Directory Manager
- Setting System Operational Attributes by Using ldapmodify

See Also: "Kinds of Attribute Information" on page 2-5

Setting System Operational Attributes by Using Oracle Directory Manager

You can view and set some of the operational attributes for each Oracle directory server to which you are connected by using **Oracle Directory Manager**. To do this, in the navigator pane, expand Oracle Internet Directory Servers, then select a server. System operational attributes appear in the right pane.

The next table describes the fields displayed in Oracle Directory Manager for each system operational attribute.

Field	Description	Default Value	Modifiable?
Configuration Set Location	DN of the entry holding the top of the naming context in this server	`cn=subconfigsubentry`	No
Indexed Attribute Locations	DN for the file containing all indexed attributes	`cn=catalogs`	No
Naming Contexts	DN for the naming contexts contained in this server. Enter a new value in the field. If you are not sure of the value, click Browse to bring up a search window.	none	Yes
Oracle Directory Version	The version or release of Oracle Internet Directory that you are using	2.1.1.0.0	No
Password Encryption	Hash algorithm for encrypting the password. Options are: - **MD4** - **MD5** - No encryption - **SHA** - **UNIX Crypt**	MD4	Yes
Process Instance Location	DN of the entry holding the Instance Registry in this server	`cn=subschemasubentry`	No
Query Entry Return Limit	Maximum number of entries to be returned by a search	1000	Yes
Replication Agreements	DN of the entry holding the replication agreement	`cn=orclareplagreements`	No
Replication Log Location	DN of the entry holding the change log in this server	`cn=changelog`	No
Replication Status Location	DN of the entry holding the change status in this server	`cn=changestatus`	No
Schema Definition Location	DN of the schema	`cn=subschemasubentry`	No

Field	Description	Default Value	Modifiable?
Server Mode	Determines whether data can be written to the server. You can change this value to either Read / Write or Read Only. Change the default to Read Only during replication process.	Read / Write	Choices are Read / Write and Read-Only
Server Operation Time Limit	Maximum amount of time, in seconds, allowed for a search to be completed	3600	Yes
Supported Control	Extension information for any LDAP operation. The control types supported by Oracle Internet Directory are listed as values of the `supportedcontrol` attribute in the root DSE. Each control type has an associated object identifier defined by the LDAP standard.The values of the supportedcontrol attribute are standard object identifiers assigned to control types.	`manageDSACtrl`	No

Setting System Operational Attributes by Using ldapmodify

The modifiable system operational attributes are:

Attribute	Description	Default
`namingContexts`	Topmost DNs for the naming contexts contained in this server. You must have super user privileges to publish a DN as a naming context.	none
`orclCryptoScheme`	Hash algorithm for encrypting the password. Options are: ■ MD4 ■ MD5 ■ No encryption ■ SHA ■ UNIX Crypt	MD4
`orclSizeLimit`	Maximum number of entries to be returned by a search	1000

Attribute	Description	Default
`orclServerMode`	Determines whether data can be written to the server. Change the default to Read-Only during replication process.	Read / Write
`orclTimeLimit`	Maximum amount of time, in seconds, allowed for a search to be completed	3600

See Also: "ldapmodify Syntax" on page A-15 for a more detailed discussion of ldapmodify, and a list of its options

Managing Naming Contexts

To enable users to search for specific naming contexts, you can publish those naming contexts. To do this, you specify the topmost entry of each naming context as a value of the `namingContexts` attribute in the root DSE.

For example, suppose you have a DIT with three major naming contexts, the topmost entries of which are `c=uk`, `c=us`, and `c=de`. If these entries are specified as values in the `namingContexts` attribute, then a user, by specifying the appropriate filter, can find information about them by searching the root DSE. The user can then focus the search—for example, by concentrating on the `c=de` naming context in particular.

To publish a naming context, you can use either Oracle Directory Manager or ldapmodify. The `namingContexts` attribute is multi-valued, so you can specify multiple naming contexts.

To search for published naming contexts, perform a base search on the root DSE with `objectClass` =* specified as a search filter. The retrieved information includes those entries specified inWthe `namingContexts` attribute.

Before you publish a naming context, be sure that:

- You are a directory administrator with the necessary access to the root DSE
- The topmost entry of that naming context exists in the directory

This section contains these topics:

- Publishing Naming Contexts by Using Oracle Directory Manager
- Publishing Naming Contexts by Using ldapmodify

Publishing Naming Contexts by Using Oracle Directory Manager

1. In the navigator pane, expand Oracle Internet Directory Servers and select the directory server on which you want to specify a naming context. The corresponding tab pages for that directory server appear in the right pane.
2. In the System Operational Attributes tab page, in the Naming Contexts field, enter the topmost DN of the naming context you want to publish. You can also click Browse to open a search window.
3. Click Apply.

Publishing Naming Contexts by Using ldapmodify

The following example input file specifies the entry `c=uk` as a naming context.

```
dn:
changetype: modify
add: namingcontexts
namingcontexts: c=uk
```

Managing Passwords

This section contains these topics:

- Managing Password Policies
- Managing Password Protection

Managing Password Policies

A password policy is a set of rules that govern how passwords are used. When a user attempts to bind to the directory, the directory server uses the password policy to ensure that the password meets the requirements set in that policy.

You establish a password policy by assigning values to the following attributes:

Policy	Attribute	Description
Lockout Duration	`pwdLockoutDuration`	The number of seconds a user is locked out of the directory if *both* of the following are true: ■ Account lockout is enabled ■ The user has been unable to bind successfully to the directory for at least the number of times specified by `pwdMaxFailure` You can set user lockout for a specific duration, or until the administrator resets the user's password. A default value of `0` (zero) means that the user is locked out forever.
Password Expiration Warning	`pwdExpireWarning`	The length of time before password expiration that the directory server sends the user a warning. If password expiration is enabled, then, by default, the directory server sends the user a warning three days before the password expires. The directory server sends the warning at each logon. If the user does not modify the password before it expires, then the directory server enforces the modification. This means that the user is locked out until the password is changed by the administrator. For this feature to work, the client application must support it.
Password Failure Count Interval	`pwdFailureCountInterval`	The number of seconds after which the password failure times are purged from the user entry. If this attribute is not present, or if it has a value of `0` (zero), then failure times are never purged.
Password Lockout	`pwdLockout`	Specification for whether users are locked out of the directory after the number of consecutive failed bind attempts specified by `pwdmaxFailure`. If the value of this policy attribute is TRUE, then users are locked out. If this attribute is not present, or if the value is FALSE, then user are not locked out and the value of `pwdMaxFailure` is ignored. By default, no account lockout is enforced.
Password Maximum Age	`pwdMaxAge`	The maximum length of time, in seconds, that a given password is valid. If this attribute is not present, or if the value is `0` (zero), then the password does not expire. By default, user passwords never expire.

Policy	Attribute	Description
Password Maximum Failure	`pwdMaxFailure`	The number of consecutive failed bind attempts after which a user account is locked. If this attribute is not present, or if the value is `0` (zero), then the account is not locked due to failed bind attempts, and the value of the password lockout policy is ignored.

Note: All user passwords are assumed to be single-valued, as mentioned in the July 2000 version of the IETF draft: `http://ietf.org/internet-drafts/draft-behera-ldap-password-policy-03.txt`

To establish a password policy, you use these two auxiliary object classes:

`pwdPolicy` Container for password policy information for the entire directory. You set these values during installation. An entry of this object class is created during installation. It has this DN: `cn=pwdpolicyentry,cn=`Oracle Internet Directory. In release 3.0.1, the policy specified applies to the entire directory.

This object class contains these attributes:

- `pwdMaxAge`
- `pwdLockout`
- `pwdLockoutDuration`
- `pwdMaxFailure`
- `pwdFailureCountInterval`

The default value for each of these attributes is 0 (zero).

`pwdInfObject` Container for password policy state for each user. This object class contains these attributes:

- `pwdChangetime`: The timestamp of the user password creation or modification
- `pwdExpirationWarned`: The time at which the first password expiration warning is been sent to the user
- `pwdFailuretime`: The timestamp of consecutive failed login attempts by the user
- `pwdAccountLockedTime`: The time at which the user account was locked

See Also: The July 2000 version of the following IETF draft: `http://ietf.org/internet-drafts/draft-behera-ldap-password-policy-03.txt`

Setting Password Policies by Using Oracle Directory Manager

To set password policies by using Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory Servers, then select the directory server instance. The corresponding tab pages appear in the right pane.
2. In the right pane, select the Password Management tab.
3. In the Account Lockout field, enter 1 to enable, or 0 to disable, account lockout.
4. In the Account Lockout Duration field, enter the number of seconds a user is locked out of the directory if *both* of the following are true:

   - Account lockout is enabled
   - The user has been unable to bind successfully to the directory for at least the number of times specified by `pwdMaxFailure`

   You can set user lockout for a specific duration, or until the administrator resets the user’s password. A default value of 0 (zero) means that the user is locked out forever.
5. In the Password Maximum Failure field, enter the number of consecutive failed bind attempts after which a user account is locked.
6. In the Password Failure Count Interval field, enter the number of seconds after which the password failure times are purged from the user entry.

7. In the Password Expiry Time field, enter the number of seconds that a given password is valid. If this attribute is not present, or if the value is 0, then the password does not expire. By default, user passwords never expire.

Setting Password Policies by Using Command-Line Tools

The following example enables the `pwdLockout` attribute, changing it from its default setting of 0 (zero).

The file `my_file.ldif` contains:

```
dn:cn=pwdpolicyentry,cn=Oracle Internet Directory
changetype:modify
replace: pwdlockout
pwdlockout: 1
```

The following command loads this file into the directory:

```
ldapmodify -p 389 -h myhost -f my_file.ldif
```

Managing Password Protection

During installation, you were prompted to set the one-way hashing scheme for maintaining password protection. Options presented to you were:

- **MD4** —A one-way hash function that produces a 128-bit hash, or message digest
- **MD5**—An improved, and more complex, version of MD4
- **SHA**—Secure Hash Algorithm, which produces a 160-bit hash, longer than MD5. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.
- **UNIX Crypt**—The UNIX hashing algorithm
- No Hashing

The hashing algorithm value you specified at installation is stored in the `orclCryptoScheme` attribute in the **root DSE**. You can change that value by using either Oracle Directory Manager or ldapmodify. You must be a superuser to do this.

Managing Password Protection by Using Oracle Directory Manager

To change the type of password protection by using Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory Servers and select the directory server instance for which you want to reset password hashing. The corresponding tab pages for that directory server appear in the right pane.
2. In the System Operational Attributes tab page, in the Password Encryption field, select the type of password hashing you want to use. Options are:

- MD4
- MD5
- No encryption
- SHA
- UNIX Crypt

3. Click Apply.

Managing Password Protection by Using ldapmodify

The following example changes the password hashing algorithm to SHA by using an LDIF file named `my_ldif_file`:

```
ldapmodify -h myhost -p 389 -v -f my_ldif_file
```

The LDIF file, `my_ldif_file`, contains:

```
dn:
changetype: modify
replace: orclcryptoscheme
orclcryptoscheme: SHA
```

See Also: "Password Protection" on page 11-6

Managing Super Users, Guest Users, and Proxy Users

A **super user** is a special directory administrator who typically has full access to directory information. The default user name of the super user is `orcladmin`; the default password is `welcome`. Oracle Corporation recommends that you change the password immediately.

A **guest user** is one who is not an anonymous user, and, at the same time, does not have a specific user entry. The default user name for a guest user is `guest`; the default password is `guest`.

A **proxy user**, as described in "Authentication Through a Middle Tier" on page 11-3, is typically used in an environment with a middle tier such as a firewall, a RADIUS server, or an LDAP self-service servlet. The default user name for a proxy user is `proxy`; the default password is `proxy`.

You can administer user names and passwords for the super, guest, and proxy users by using either Oracle Directory Manager or ldapmodify.

> **Note:** It is possible to log on to the Oracle Directory Manager without giving a user name or password. If you do this, you have the privileges specified for an anonymous user. Anonymous users should have very limited privileges.

> **See Also:** Chapter 13, "Managing Directory Access Control" for information on how to set access rights

This section contains these topics:

- Managing Super, Guest, and Proxy Users by Using Oracle Directory Manager
- Managing Super, Guest, and Proxy Users by Using ldapmodify

Managing Super, Guest, and Proxy Users by Using Oracle Directory Manager

> **Note:** The passwords for superusers, guest users, and proxy users are encrypted by default. You cannot modify them to send them in the clear.

To set a user name or password for a super user, a guest user, or a proxy user by using Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory Servers.
2. Select a server. The group of tab pages for that server appear in the right pane.

3. Select the System Passwords tab. This page displays the current user names and passwords for each type of user. Note that passwords are not displayed in the password fields.

   The next table lists and describes the fields in the System Passwords tab page.

Field	Description
Super User Name	Type the super user name. The default is `orcladmin`.
Super User Password	Type the super user password. The default is `welcome`. You should change this password immediately.
Guest Login Name	Type the guest login name. Guests have privileges determined by the **access control policy point**s (**ACPs**) in the directory. The default is `guest`.
Guest Login Password	Type the guest login password. The default is `guest`.
Proxy Login Name	Type the proxy login name. Proxy users have privileges determined by the ACPs in the directory. The default is `proxy`.
Proxy Login Password	Type the proxy login password. The default is `proxy`. You should change this password immediately.

4. Edit the appropriate field in the System Passwords tab page. To save your changes, click Apply.

Managing Super, Guest, and Proxy Users by Using ldapmodify

To set or modify a user name or password for a superuser, a guest user, or a proxy user, use ldapmodify to modify the appropriate attribute:

User Name/Password	Attribute
Super user name	`orclsuname`
Super user password	`orclsupassword`
Guest user name	`orclguname`
Guest user password	`orclgupassword`
Proxy user name	`orclprname`
Proxy user password	`orclprpassword`

For example, to change the password of the super user to *superuserpassword*, use ldapmodify to modify the **directory-specific entry (DSE)** by using an LDIF file containing the following:

```
dn:
changetype:modify
replace:orclsupassword
orclsupassword:superuserpassword
```

Configuring Searches

See Also: "ldapmodify Syntax" on page A-15 for ldapmodify syntax and usage notes.

You can set the maximum number of entries returned in searches, as well as the maximum amount of time, in seconds, for searches to be completed. You can do both of these by using either Oracle Directory Manager or ldapmodify.

This section contains these topics:

- Configuring Searches by Using Oracle Directory Manager
- Configuring Searches by Using ldapmodify

Configuring Searches by Using Oracle Directory Manager

You can use Oracle Directory Manager to set the maximum number of retries returned in searches and the maximum amount of time to allow for searches.

Setting the Maximum Number of Entries Returned in Searches by Using Oracle Directory Manager

1. In the navigator pane, expand Oracle Internet Directory Servers and select a directory server instance. The group of tab pages for that server appear in the right pane.
2. In the System Operational Attributes tab page, in the Query Entry Return Limit field, enter the maximum number of entries to be returned by a search. The default is 1000.
3. Click Apply.

Setting the Maximum Amount of Time For Searches by Using Oracle Directory Manager

1. In the navigator pane, expand Oracle Internet Directory Servers and select a directory server instance. The group of tab pages for that server appear in the right pane.
2. In the System Operational Attributes tab page, in the Server Operation Time Limit, enter the maximum number of seconds for a search to be completed. The default is 3600.
3. Click Apply.

Configuring Searches by Using ldapmodify

You can use ldamodify to set the maximum number of retries returned in searches and the maximum amount of time to allow for searches.

Setting the Maximum Number of Entries Returned in Searches by Using ldapmodify

The following example changes the maximum number of entries to be returned in searches to 500.

```
ldapmodify -h myhost -p 389 -v <<EOF
dn:
changetype: modify
replace: orclsizelimit
orclsizelimit: 500
EOF
```

Setting the Maximum Amount of Time For Searches by Using ldapmodify

The following example changes the maximum amount of time for a search to 2400.

```
ldapmodify -h myhost -p 389 -v <<EOF
dn:
changetype: modify
replace: orcltimelimit
orcltimelimit: 2400
EOF
```

See Also: "ldapmodify Syntax" on page A-15

Setting Debug Logging Levels

You can set debug logging levels by using either **Oracle Directory Manager** or the **OID Control Utility**.

This section contains these topics:

- Setting Debug Logging Levels by Using Oracle Directory Manager
- Setting Debug Logging Levels by Using the OID Control Utility

Setting Debug Logging Levels by Using Oracle Directory Manager

To set the debug logging level:

1. In the navigator pane, expand Oracle Internet Directory Servers and select a server instance. The group of tab pages for that server appear in the right pane.
2. Select the Debug Flags tab.

   Ordinarily, you can leave the check boxes on this tab page unselected. However, to generate a log for a specific problem, use this tab page to specify the debug logging level.

Setting Debug Logging Levels by Using the OID Control Utility

To set debug logging levels by using the OID Control Utility, restart the Oracle directory server using the `-debug` flag for an LDAP server, and the `-d` flag for the replication server. Use the debug level number based on Table 6–1.

Because debug levels are additive, you need to sum together the numbers representing the functions that you want to activate, and use that sum in the command-line option.

By default, debug logging is turned off. To turn it on, modify the **directory-specific entry (DSE)** attribute `orcldebugflag` to the level you want. You can configure debug levels to one of the following levels.

To see debug log files generated by the OID Control Utility, navigate to `$`*`ORACLE_HOME`*`/ldap/log`.

Table 6–1 provides the complete list of debug logging levels.

Table 6–1 Debug Logging Levels

Logging Level Value	Function
1	Trace function calls
2	Debug packet handling
4	Heavy trace debugging
8	Connection management
16	Print out packets sent and received
32	Search filter processing
64	Configuration file processing
128	Access control list processing
256	Stats log connections/operations/results
512	Stats log entries sent
1024	Print communication with the back-end
2048	Print entry parsing debugging
4096	Schema-related debugging
32768	Replication-specific debugging
65535	Enable all debugging

For example, to trace function calls (1) and active connection management (8), enter 9 as the debug level (8 + 1 = 9) as follows:

```
oidctl server=oidldapd instance=1 flags='-debug 9' restart
oidctl server=oidrepld instance=1 flags='-h my_host -p 389 -d 9' restart
```

This example restarts both the Oracle directory server as well as the Oracle directory replication server with the debugging flags.

Using Audit Log

The audit log records critical events on the Oracle directory server that are important from both a security and an operational point of view. An administrator can query the audit log by using ldapsearch commands. Because the log generation is contingent upon events occurring on the server, only the Oracle directory server itself can create the log entries. You cannot add audit log entries by using either **Oracle Directory Manager** or the command-line tools.

The audit log is made up of regular directory entries, one entry for each event. You can specify search criteria using ldapsearch, and you can view the audit log entries by using Oracle Directory Manager.

By default, audit logging is turned off. To turn it on, modify the **directory-specific entry (DSE)** attribute `orclauditlevel` to the level you want. You can configure audit levels to audit selected events only.

See Also:

- "Auditable Events" on page 6-31 for a listing of audit levels
- "Searching for Audit Log Entries by Using Oracle Directory Manager" on page 8-6
- "Searching for Audit Log Entries by Using ldapsearch" on page 6-34
- "bulkdelete Syntax" on page A-27

This section contains these topics:

- Structure of Audit Log Entries
- Position of Audit Log Entries in the DIT
- Auditable Events
- Setting the Audit Level
- Searching for Audit Log Entries
- Purging the Audit Log

Structure of Audit Log Entries

Each audit log entry contains the orclAuditoc **object class**. Like all other structural object classes, orclAuditoc inherits from top. Its attributes include:

Attribute	Description
orclsequence	Used to create the name of the entry. The name is generated using a database sequence.
orcleventtype	Specifies the type of event that occurred. This is a cataloged attribute.
orcleventtime	Specifies the time at which the event occurred. This is formatted in **UTC (Coordinated Universal Time)**. UTC is indicated by a z at the end of the value. For example, orcleventtime: 199811281010z
orcluserdn	Specifies the identity of the user who logged into the Oracle directory server to perform the operation. This attribute is catalogued.
orclopresult	Specifies the outcome of the operation. It states either SUCCESS if the operation succeeds, or the reason why the operation failed.
orclauditmessage	Specifies the textual message. This attribute is not catalogued.
objectclass	Contains the preset values top and orclauditoc.

Note that the audit log entries do not become part of a regular search result set even though the search filter can satisfy the query criteria. For example, a search with the condition objectclass=top does not yield results from the auditlog entries. Only a search with cn=auditlog as the base of the search can find audit log entries.

Note: By default, the attributes orcleventtype and orcluserdn are indexed at installation of Oracle Internet Directory. If you drop the indexes from these attributes, you cannot search for them. To re-create the index for these attributes, use the Catalog Management tool. See "Indexing an Attribute by Using Command-Line Tools" on page 7-30.

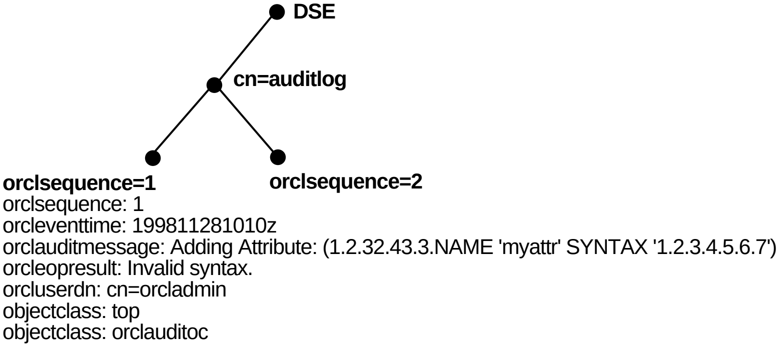
See Also:

- "Catalog Management Tool Syntax" on page A-32 for information about catalogued attributes
- "Object Class Types" on page 2-10 for a description of `top`

Position of Audit Log Entries in the DIT

The audit log container is part of the DSE. It holds its entries as children, organized according to the `orclsequence` attribute. See Figure 6–2.

Figure 6–2 Sample Audit Log in DSE



```
orclsequence: 1
orcleventtime: 199811281010z
orclauditmessage: Adding Attribute: (1.2.32.43.3.NAME 'myattr' SYNTAX '1.2.3.4.5.6.7')
orcleopresult: Invalid syntax.
orcluserdn: cn=orcladmin
objectclass: top
objectclass: orclauditoc
```

Auditable Events

The next table shows the auditable events and their audit levels. The third column, Audit Levels, contains hexidecimal values. You can audit more than one event by adding their corresponding values found in this column.

Table 6–2 Auditable Events

Event	Description	Audit Levels
Superuser login	Super user bind to the server (successes or failures)	0x0001
Schema element add / replace	Addition of a new schema element (successes or failures)	0x0002
Schema element delete	Deletion of a schema (successes or failures)	0x0004

Table 6–2 Auditable Events

Event	Description	Audit Levels
Bind	Unsuccessful bind cases	0x0008
Access violation	Access denied by **access control policy point**	0x0010
directory-specific entry (DSE) modification	Changes to a **directory-specific entry (DSE)** (successes or failures)	0x0020
Replication login	Replication server authentication (successes or failures)	0x0040
ACL modification	Changes to an **access control list (ACL)**	0x0080
User password modification	Modification of user password attribute	0x0100
Add	ldapadd operation (successes or failures)	0x0200
Delete	ldapdelete operation (successes or failures)	0x0400
Modify	ldapmodify operation (successes or failures)	0x0800
ModifyDN	ldapModifyDN operation (successes or failures)	0x1000

Setting the Audit Level

Events described in the previous section can be turned on or off. The DSE attribute `orclauditlevel` indicates the current audit level set on the server. A value of 0 for the attribute means no auditing, which is the default.

You can set the audit level by using either Oracle Directory Manager or ldapmodify. Both methods are described in this section.

Setting the Audit Level by Using Oracle Directory Manager

To set the audit level by using Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory Servers and select the directory server instance.
2. In the right pane, select the Audit Mask Levels tab page.
3. Select the check box for the audit level you want to use.
4. Click Apply.

Both successful and uneventful events are entered into the audit log if they are selected, except:

- Bind, which logs only unsuccessful bind attempts
- Access Violation, which logs only events in which access is denied by an ACP.

Note: Remember: The changes will not affect the active directory server instance until you restart it. See "Restarting Directory Server Instances" on page 4-7.

See Also: "Auditable Events" on page 6-31 for a description of each audit level

Setting the Audit Level by Using ldapmodify

To audit more than one event, add the values of their the audit masks. For example, suppose you want to audit the following three events:

Event	Audit Level	Value
Schema element delete	0x0004	4
DSE modification	0x0020	32
Add	0x0200	512
Total		548

The total value of the audit levels is 548. The ldapmodify command would therefore look something like this:

```
ldapmodify -p port -h host << EOF
dn:
changetype:modify
replace: orclauditlevel
orclauditlevel: 548
EOF
```

Restart the directory server instance after any changes are made to `orclauditlevel` for the changes to take effect.

See Also: "Task 3: Reset the Default Security Configuration" on page 4-9

Searching for Audit Log Entries

You can search for audit log entries by using either Oracle Directory Manager or ldapsearch.

Searching for Audit Log Entries by Using Oracle Directory Manager

> **See:** "Searching for Audit Log Entries by Using Oracle Directory Manager" on page 8-6

Searching for Audit Log Entries by Using ldapsearch

The **DN** for the audit log container is `cn=auditlog`. To search for audit log entries, perform a subtree or one-level search, with the container object `cn=auditlog` as the base of the search.

> **See:** "ldapsearch Syntax" on page A-22

Purging the Audit Log

You can use bulkdelete to purge audit log objects under the container `cn=auditlog`. Run the following command:

```
bulkdelete.sh -connect net_service_name -base "cn=auditlog"
```

Viewing Active Server Instance Information

You can use **Oracle Directory Manager** to view information about any active directory server instance. To do this:

1. In the navigator pane, expand Oracle Internet Directory Servers and select a directory server. The group of tab pages for that directory server instance appear in the right pane.
2. Select the Server Management tab. This displays basic information—namely, type, instance number, debug level, and host name—for all active directory server instances.
3. To see configuration parameters for a particular directory server instance, select the directory server instance, then click View Properties. The Server Process dialog box displays configuration parameters for the directory server instance you selected. Note that you cannot change configuration parameters in this dialog box. To change them, you must change the configuration set entry on which they are based.

See Also: "Managing Server Configuration Set Entries by Using Oracle Directory Manager" on page 6-4 for instructions on changing configuration set entries

Changing the Password to an Oracle Database Server

The Oracle Internet Directory uses a password when connecting to an Oracle database. The default for this password when you install Oracle Internet Directory is ODS. You can change this password by using the **OID Database Password Utility**.

See Also: "OID Database Password Utility Syntax" on page A-41

7

Managing the Directory Schema

This chapter explains how to administer the Oracle Internet Directory object classes and attributes.

This chapter contains these topics:

About the Directory Schema

A directory schema does the following:

- Contains rules about the kinds of objects you can store in the directory
- Contains rules for how directory servers and clients treat information during operations such as a search
- Helps to maintain the integrity and quality of the data stored in the directory
- Reduces duplication of data
- Provides a predictable way for directory-enabled applications to access and modify directory objects

The directory schema contains all information about how data is organized in the DIT. It includes attribute types, and the syntaxes and matching rules that apply to them. It also contains the various groupings of attributes, called object classes.

This chapter discusses each of these elements.

See Also: "The Directory Schema" on page 2-13

About Object Class Management

This section explains how to add and modify an **object class**. Oracle Corporation recommends that you understand the basic concepts of directory components before attempting to add to or modify the base schema in the directory.

See Also:

- "Object Classes" on page 2-8 for a conceptual overview of object classes
- Appendix C, "Schema Elements" for a list of schema components installed with Oracle Internet Directory

This section contains these topics:

- Guidelines for Adding Object Classes
- Guidelines for Modifying Object Classes
- Guidelines for Deleting Object Classes

Guidelines for Adding Object Classes

When you add directory entries, you select object classes for those entries. The attributes of an entry are determined by the object classes to which that entry is assigned.

Entries must be loaded in a top-down sequence. When you add an entry, all of its parent entries must already exist in the directory. Similarly, when you add entries that reference object classes and attributes, those referenced object classes and attributes must already exist in the directory schema. In most cases this will not be a problem since the directory server is delivered with a full set of standard directory objects.

Note: Every schema object in the Oracle Internet Directory has certain limitations. For example, some objects cannot be changed. These limitations are explained as constraints and rules in this chapter.

The attributes that an entry **inherits** from an object class may be either mandatory or optional. Optional attributes need not be present in the directory entry.

You can specify for any object class whether an attribute is mandatory or optional; however, the characteristic you specify is binding only for that object class. If you place the attribute in another object class, you can again specify whether the attribute is mandatory or optional for that object class. You can:

- Select from existing standard object classes
- Add a new, non-standard object class and assign it existing attributes
- Modify an existing object class, assigning it a different set of attributes
- Add and modify existing attributes

See Also: "About Attribute Management" on page 7-16

Administrators typically assign object classes to entries based on the attributes present in that object class. However, **superclasses** let you take advantage of inheritance—that is, the object classes selected for an entry have a hierarchy of superclasses from which they inherit mandatory and optional attributes. By default, all object classes inherit from the `top` object class.

When you add or perform an operation on an entry, you do not need to specify the entire hierarchy of superclasses associated with that entry. This feature, called object class explosion, enables you to specify only the leaf object classes. Oracle Internet Directory resolves the hierarchy for the leaf object classes and enforces the information model constraints. For example, the `inetOrgPerson` object class has `top`, `person` and `organizationalPerson` as its superclasses. When you create an entry for a person entry, you need to specify only `inetOrgPerson` as the object class. Oracle Internet Directory then enforces the schema constraints defined by the respective superclasses, namely, `top`, `person`, and `organizationalPerson`.

When you add object classes, keep the following guidelines in mind:

- Every structural object class must have `top` as a superclass.
- The name and the object identifier of an object class must be unique across all the schema components.
- Schema components referred to in the object class, such as superclasses, must already exist.
- The superclass of an abstract object class must be abstract also.
- It is possible to redefine mandatory attributes in a superclass into optional attributes in the new object class. Conversely, optional attributes in a superclass can be redefined into mandatory attributes in the new object class.

> **See Also:** "Subclasses, Superclasses, and Inheritance" on page 2-9 for a conceptual discussion of these terms

Guidelines for Modifying Object Classes

This section discusses the types of modifications you can make to an existing object class. You can perform modifications through Oracle Directory Manager and through the command-line tools.

You can make these changes to an object class:

- Change a mandatory attribute into an optional attribute
- Add optional attributes
- Add additional superclasses
- Convert *abstract* object classes into *structural* or *auxiliary* object classes unless the abstract object class is a superclass to another abstract object class

When you modify object classes, keep these guidelines in mind:

- You cannot modify an object class that is part of the standard LDAP schema. You can, however, modify user-defined object classes. Also, if existing object classes do not have the attributes you need, you can create an auxiliary object class and associate the needed attributes with it.

- You cannot add additional mandatory attributes to an existing object class.
- You cannot modify object classes in the base schema.
- You cannot remove attributes or superclasses from an existing object class.
- You cannot convert structural object classes to other object class types.
- You should not modify an object class if there are entries already associated with it.

See Also:

- "Managing Object Classes by Using Oracle Directory Manager" on page 7-6
- "Managing Object Classes by Using Command-Line Tools" on page 7-14

Guidelines for Deleting Object Classes

There are also some limitations on deleting object classes:

- You cannot delete object classes from the base schema.
- You can delete object classes that are not in the base schema as long as they are not directly or indirectly referenced by other schema components. For example, there may be some directory entries referring to these object classes. Deleting these object classes renders these entries inaccessible.

Note: Oracle Internet Directory does not enforce these rules. They are provided here as guidelines.

Managing Object Classes by Using Oracle Directory Manager

This section contains these topics:

- Searching for Object Classes by Using Oracle Directory Manager
- Viewing Properties of Object Classes by Using Oracle Directory Manager
- Adding Object Classes by Using Oracle Directory Manager
- Modifying Object Classes by Using Oracle Directory Manager
- Deleting Object Classes by Using Oracle Directory Manager

Searching for Object Classes by Using Oracle Directory Manager

You can specify your search for an object class by:

- Selecting an object class property, for example, a name or an object identifier
- Entering a value for the property you selected
- Selecting a search filter specifying the relationship between the object class property you selected and the value you entered, for example, Begins With or Exactly Matches

This section provides more details on how to enter an object class search.

To search for an object class:

1. In the navigator pane, select Schema Management. The Schema Management tab pages appear in the right pane.
2. Click the Find Object Classes button at the lower right of the right pane, or, from the menu bar, click Edit > Find Object Classes. The Find: Object Classes dialog box appears.

3. In the menu farthest to the left on the search criteria bar, select the property of the object class for which you want to search. Options are:

Option	Description
Name	Name of the object class for which you are searching. For example, the phrase `Name Exact Match subAcl` gives you the `subAcl` object class.
Object ID	Object Identifier for the object class for which you are searching. For example, the phrase `Object ID Begins With 2.5.2` gives you a list of object classes whose object identifiers begin with 2.5.2.
Description	Word in the description field. For example, the phrase `Description Contains Shoe` gives you a list of object classes with the word *`shoe`* in the description column.
Type	Type of object class for which you are searching, whether abstract, structural, or auxiliary
Superclass	Class from which the object class for which you are searching is derived
Mandatory Attributes	Mandatory attributes of the object class for which you are searching. For example, the phrase `Mandatory Attributes Contains cn` gives you a list of all object classes in which the `cn` attribute is mandatory.
Optional Attributes	Optional attributes of the object class for which you are searching

Note: Not all attributes are used in every object class. Be sure that the attribute you specify actually corresponds to one in the object class for which you are looking. Otherwise, the search will fail.

4. In the menu in the middle of the search criteria bar, select the filter you want to use for your search. Options are:

Filter	Description
Begins With	Searches by using only the first few characters of the property of the object class for which you are searching. For example, the phrase `Type Begins With aux` gives you a list of all of the auxiliary object classes.
Ends With	Searches by using only the last few characters of the property of the object class for which you are searching. For example, the phrase `Type Ends With ral` gives you a list of all of the structural object classes.

Filter	Description
Contains	Searches for object classes in which the property you selected includes, but is not necessarily limited to, the value you enter. For example, the phrase `Optional Attributes Contains cn` gives you a list of all object classes in which `cn` is an optional attribute.
Exact Match	Searches for an object class in which the property you selected is exactly the same as the value you enter. For example, the phrase `Super Class Exact Match person` gives you a list of all object classes that have `person` as their superclass.
Greater Or Equal	Searches for an object class in which the property you selected is numerically or alphabetically greater than or equal to the value you enter. For example, the phrase `Name Greater or Equal orcl` gives you a list of object classes from those beginning with the letters `orcl` to those beginning with letters at the end of the alphabet.
Less or Equal	Searches for an object class in which the property you selected is numerically or alphabetically less than or equal to the value you enter. For example, the phrase `Name Less or Equal orcl` gives you a list of object classes from those beginning with the letters `orcl` to those at the beginning of the alphabet.
Not Null	Searches for all object classes in which the property you selected is present. For example, the phrase `Mandatory Attributes Not Null` gives you a list of all object classes which contain mandatory attributes.

5. In the text box at the right end of the search criteria bar, type the value of the property of the object class for which you are searching. For example, to search for all object classes in which the name of the object class begins with the letters `orcl`, type those letters in the text box at the right end of the search criteria bar.

6. Below the Search Criteria field are five buttons described in the next table. Use these buttons to further refine your search.

Button	Description
New	Creates a new search criteria bar in the Search Criteria field. This button is enabled only when the search criteria bar has been deleted.
And	Creates another search criteria bar in the Search Criteria field. Matches all object classes having one specified criterion with those that also have another specified criterion.
Or	Creates another search criteria bar in the Search Criteria field. Matches all object classes with either one specified attribute or another.
Not	Negates the criterion in the selected search criteria bar and retrieves all object classes that do not have the specified criterion.
Delete	Deletes a selected search criteria bar

7. Click Search. The results of your search appear in the window at the lower portion of the Find:Object Class dialog box.

Viewing Properties of Object Classes by Using Oracle Directory Manager

To view all object classes in the schema:

1. In the navigator pane, expand Schema Management. The tabs in the Schema Management pane display the components of the schema:
   - Object classes
   - Attributes
   - Syntaxes
   - Matching Rules
2. In the right pane, select the Object Classes tab page.

   To examine an individual object class and its attributes, in the Object Classes tab page, click the object class. The properties of the selected object class appear in the Object Class dialog box.

3. In the Object Class dialog box:

   - Object classes from which attributes may be inherited are listed in the Super Class box
   - Mandatory attributes are listed in the Mandatory Attributes box
   - Optional attributes are listed in the Optional Attributes box

   Each box indicates whether the attributes are indexed so that they can be used in a search expression.

Adding Object Classes by Using Oracle Directory Manager

To add object classes by using Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server*, then select Schema Management.
2. Choose one of the following methods:

   - In the right pane, select the Object Classes tab and click the Create button in the toolbar.
   - Click the Create button at the bottom of the right pane.
   - From Operations menu, select Create Object Class.

   The New Object Class dialog box appears.

   Alternatively, select an object class that is similar to one you would like to create, and then click Create Like. A dialog box appears; it includes the attributes of the selected object class. You can create the new object class using the selected one as a template.

3. Enter the information in the fields described in the following table:

Field	Description
Name	Enter the name of the object class you are creating.
Object ID	Enter the object identifier. This is a standardized numerical sequence based on IETF standards. It must be unique, and should comply with the system established within your organization. Normally it is derived from the identifier assigned by registration agencies, such as ANSI or ISO.
Description	Use this optional field for your information only.
Type	Specify the type of object class: Abstract, Structural, Auxiliary, None.
Super Class	Specify the class(es) from which to derive this object class. This object class will inherit all the attributes of the superclass(es) you select. Every structural object class must have `top` as one of its superclasses. Clicking Add displays the Super Class Selector dialog box from which you can select the superclass(es) you want to add.
Mandatory Attributes	Specify the attributes for which values must be entered. Clicking Add displays the Mandatory Attributes Selector dialog box from which you can select the mandatory attributes you want to add.
Optional Attributes	Specify the attributes for which values are not required. Clicking Add displays the Optional Attributes Selector dialog box from which you can select the optional attributes you want to add.

4. Click OK.

See Also:

- "Object Class Types" on page 2-10
- "Subclasses, Superclasses, and Inheritance" on page 2-9
- Oracle Directory Manager online help for further details about adding object classes

Modifying Object Classes by Using Oracle Directory Manager

To modify an object class:

1. In the navigator pane, select Schema Management, then select the Object Classes tab.
2. In the Object Classes tab page, double-click the object class you want to modify. The Object Class dialog box appears.
3. Modify or add the information in the fields described in the following table.

Field	Description
Name	Enter the name of the object class you are creating.
Object ID	Enter the object identifier. This is a standardized numerical sequence based on IETF standards. It must be unique, and should comply with the system established within your organization. Normally it is derived from the identifier assigned by registration agencies, such as ANSI or ISO.
Description	Use this optional field for your information only.
Type	Specify the type of object class: Abstract, Structural, Auxiliary, None.
Super Class	Specify the class(es) from which to derive this object class. This object class will inherit all the attributes of the superclass(es) you select. Every structural object class must have `top` as one of its superclasses. Clicking Add displays the Super Class Selector dialog box from which you can select the superclass(es) you want to add.
Mandatory Attributes	Specify the attributes for which values must be entered. Clicking Add displays the Mandatory Attributes Selector dialog box from which you can select the mandatory attributes you want to add.
Optional Attributes	Specify the attributes for which values are not required. Clicking Add displays the Optional Attributes Selector dialog box from which you can select the optional attributes you want to add.

4. Click OK.

See Also:

- "Object Class Types" on page 2-10
- "Subclasses, Superclasses, and Inheritance" on page 2-9

Deleting Object Classes by Using Oracle Directory Manager

Caution: Oracle Corporation recommends that you not delete object classes from the schema.

Should you decide to delete an object class, be careful not to delete one that is in use or that you might want to use in the future. If you delete an object class that is referenced by any entries, those entries then become inaccessible.

Note: You can add attributes to an auxiliary object class or a user-defined structural object class.

See Also: Example: Adding a New Attribute to an Auxiliary or User-Defined Object Class on page 7-15 for an example of adding attributes to an auxiliary object class

To delete an object class by using Oracle Directory Manager:

1. In the navigator pane, select Schema Management.
2. In the right pane, select the Object Classes tab and select the object class you want to delete.
3. Click Delete.

Managing Object Classes by Using Command-Line Tools

You can use command-line tools to add or modify existing object classes in the directory schema. The command-line tools enable you to use input files. Furthermore, the commands can be batched together in scripts.

To add or modify schema components, use ldapmodify.

> **See:** "ldapmodify Syntax" on page A-15

This section contains these examples:

- Example: Adding a New Object Class
- Example: Adding a New Attribute to an Auxiliary or User-Defined Object Class

Example: Adding a New Object Class

In this example, an LDIF input file, `new_object_class.ldi`, contains data similar to this:

```
dn: cn=subschemasubentry
changetype: modify
add: objectclasses
objectclasses: ( 1.2.3.4.5 NAME ’myobjclass’ SUP top STRUCTURAL MUST ( cn $
sn ) MAY ( telephonenumber $ givenname $ myattr ) )
```

Be sure to leave the mandatory space between the opening and closing parentheses and the object identifier.

To load the file, enter this command:

```
ldapmodify –h myhost -p 389 -f new_object_class.ldi
```

This example adds the *structural* object class named myobjclass, giving it an object identifier of 1.2.3.4.5, specifying top as its superclass, requiring cn and sn as mandatory attributes, and allowing telephonenumber, givenname, and myattr as optional attributes. Note that all the attributes mentioned must exist prior to the execution of the command.

To create an *abstract* object class, follow the above example, replacing the word STRUCTURAL with the word ABSTRACT.

Example: Adding a New Attribute to an Auxiliary or User-Defined Object Class

To add a new attribute to either an auxiliary object class or a user-defined structural object class, use ldapmodify. This example deletes the old object class definition and adds the new definition in a compound modify operation. The change is committed by the Oracle directory server in one transaction. Existing data is not affected. The input file should be as follows:

```
dn: cn=subschemasubentry
changetype: modify
delete: objectclasses
objectclasses: old value
-
add: objectclasses
objectclasses: new value
```

For example, to add the attribute changes to the existing object class country, the input file would be:

```
dn: cn=subschemasubentry
changetype: modify
delete: objectclasses
objectclasses:  ( 2.5.6.2 NAME ’country’ SUP top STRUCTURAL MUST c MAY
( searchGuide $ description  )  )
-
add: objectclasses
objectclasses:  ( 2.5.6.2 NAME ’country’ SUP top STRUCTURAL MUST c MAY
( searchGuide $ description  $ changes )  )
```

About Attribute Management

This section contains these topics:

- Rules for Adding Attributes
- Rules for Modifying Attributes
- Rules for Deleting Attributes

You need to understand attributes from a conceptual standpoint before attempting operations involving attributes.

In most cases, the attributes available in the base schema will suit the needs of your organization. However, if you decide to use an attribute not available in the base schema, you can add a new attribute or modify an existing one.

By default, attributes are multi-valued. You can specify an attribute as single-valued by using either Oracle Directory Manager or command-line tools.

> **See Also:** "Attributes" on page 2-3 for a conceptual discussion of attributes

Rules for Adding Attributes

The rules for adding attributes are:

- The name and the object identifier of an attribute must be unique across all the schema components.
- Syntax and matching rules must agree.
- Any super attributes must already exist.

Rules for Modifying Attributes

The rules for modifying attributes are:

- The name and the object identifier of an attribute must be unique across all the schema components.
- The syntax of an attribute cannot be modified.
- A single-valued attribute can be made into multi-valued, but a multi-valued attribute cannot be made single-valued.
- You cannot modify or delete base schema attributes.

Rules for Deleting Attributes

The rules for deleting attributes are:

- Attributes from the base schema cannot be deleted.
- You can delete any attribute that is not referenced directly or indirectly by some other schema component.

  If you delete an attribute that is referenced by any entry, that entry will no longer be available for directory operations.

Managing Attributes by Using Oracle Directory Manager

This section contains these topics:

- Viewing All Directory Attributes by Using Oracle Directory Manager
- Searching for Attributes by Using Oracle Directory Manager
- Adding an Attribute by Using Oracle Directory Manager
- Modifying an Attribute by Using Oracle Directory Manager
- Deleting an Attribute by Using Oracle Directory Manager
- Indexing an Attribute by Using Oracle Directory Manager

> **See Also:**
>
> - "Attribute Options" on page 2-7 for information about attribute options
> - "Managing Entries with Attribute Options by Using Oracle Directory Manager" on page 8-13 and "Managing Entries with Attribute Options by Using Command-Line Tools" on page 8-17 for instructions on adding and deleting attribute options and for searching for entries containing attribute options

Viewing All Directory Attributes by Using Oracle Directory Manager

To view attributes by using Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server instance*, then select Schema Management.
2. In the right pane, select the Attributes tab. This tab page displays a table containing the attribute properties. The following table describes each column of the table in the Attributes tab page.

Column	Description
Name	The standardized attribute type names
Indexed	Check boxes indicating whether attributes are indexed
Object ID	Standardized object identifier for each attribute
Description	Words describing various attributes
Syntax	The standardized rules for data entry applicable to each attribute type
Size	Maximum size allowed for each object
Usage	Standards specifying how the attribute can be used. There are four options: `userApplications`, `directoryOperation`, `distributedOperation`, and `dSAOperation`.
Ordering	Standards specifying how precedence is established for values
Equality	Standards specifying how equality is determined in compare and search operations
Substring	Used for regular expression matching
Single Value	Indicates attribute types that contain a maximum of one value
Super	Super attribute for each attribute

> **See Also:** "Viewing Attributes for a Specific Entry by Using Oracle Directory Manager" on page 8-8 for instructions about how to view attributes for a specific entry

Searching for Attributes by Using Oracle Directory Manager

To search for attributes by using Oracle Directory Manager:

1. In the navigator pane, select Schema Management. The Schema Management tab pages appear in the right pane.
2. Select the Attributes tab page.
3. Click the Find Attributes button in the lower right corner. The Find Attributes dialog box appears
4. In the menu at the left end of the search criteria bar, select the property of the attributes for which you want to search. Options are:

Field	Description
Name	Name of the attribute for which you are searching
Indexed	List of indexed attributes
Object ID	Object Identifier for the attribute for which you are searching. For example, the phrase `Object ID Begins With 2.5.2` gives you a list of attributes whose object identifiers begin with `2.5.2`.
Description	Words in the description column of attributes
Syntax	The standardized rules for data entry applicable to this attribute type. Use this to narrow your search to attributes using a particular syntax.
Size	Maximum size allowed for this object
Usage	Standards specifying how the attribute can be used. You narrow your search by entering one of the following options: `userApplications`, `directoryOperation`, `distributedOperation`, and `dSAOperation`.
Ordering	Standards specifying how precedence is established for values
Equality	Standards specifying how equality is determined in compare and search operations
Substring	Used for regular expression matching
Single Value	Indicator that this attribute type contains a maximum of one value
Super	Super attribute for the attribute for which you are searching

5. In the menu in the middle of the search criteria bar, select the filter you want to use for your search. Options are:

Option	Description
Begins With	Searches by using only the first few characters of the property's value. For example, the phrase `Syntax Begins With 1.3` gives you a list of all attributes in which the first few numbers of the syntax identifier are *1.3*.
Ends With	Searches by using only the last few characters of the property's value. For example, the phrase `Name Ends With License` gives you a list of all attributes with that ending, such as `carLicense`.
Contains	Searches for attributes that include the property with the value you enter. For example, the phrase `Ordering Contains time` gives you a list of all attributes with the word `time` in the Ordering column.
Exact Match	Searches for a value that is exactly the same as that found in the attribute property you specified. For example, the phrase `Equality Exact Match caseIgnoreMatch` gives you a list of all attributes that have the `caseIgnoreMatch` matching rule.
Greater or Equal	Searches for an attribute that has a property that is numerically or alphabetically greater than or equal to the value you enter. For example, the phrase `Name Greater or Equal orcl` gives you a list of attributes from those beginning with `orcl` to those beginning with letters at the end of the alphabet.
Less or Equal	Searches for an attribute that has a property that is numerically or alphabetically less than or equal to the value you enter. For example, the phrase `Name Less or Equal orcl` gives you a list of attributes from those beginning with `orcl` to those beginning with letters at the start of the alphabet.
Not Null	Searches for all attributes in which the attribute property you selected is present. For example, the phrase `Description Not Null` gives you a list of all attributes which have text in the description field.

6. In the text box at the right end of the search criteria bar, type part or all of the value of the attribute for which you want to search. For example, to search for all attributes whose names begin with the letters `orcl`, you would type those letters in the text box at the right end of the search criteria bar and create the phrase `Name Begins With orcl`.

7. Beneath the Search Criteria field are five buttons described in the following table. Use these buttons to further refine your search.

Button	Description
New	Creates a new search criteria bar in the Search Criteria field. This button is enabled only when the Search Criteria field is empty.
And	Creates another search criteria bar in the Search Criteria field. Matches all attributes with one specified property with those that also have another specified property.
Or	Creates another search criteria bar in the Search Criteria field. Matches all attributes with either one specified property or another.
Not	Negates the criteria in the selected search criteria bar and matches all attributes that do not have the property specified.
Delete	Deletes a selected search criteria bar

8. Click Search. The results of your search appear in the window at the lower portion of the Find: Attributes dialog box.

Adding an Attribute by Using Oracle Directory Manager

You can add a completely new attribute, or copy from an existing one.

> **Tip:** Because equality, syntax, and matching rules are numerous and complex, it may be simpler to copy these characteristics from a similar existing attribute.

Adding a New Attribute by Using Oracle Directory Manager

To add a new attribute:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server*, then select Schema Management.

2. Do one of the following:

- In the right pane, select the Attributes tab, then click the Create button in the toolbar.
- In the right pane, select the Attributes tab, then click the Create button at the bottom of the Attributes tab page.
- From the Operation menu, select Create Attribute. The New Attribute Type dialog box appears. It contains two tab pages—General and Advanced—with fields in which you either enter values or select from menus.

3. In the General tab, enter values in each of the fields as described in the following table:

Field	Description
Name	Type the name for this attribute.
Object ID	Type the Object ID for this attribute. The Object ID is a standardized numerical sequence based on IETF standards. It must be unique. Normally this is derived from the identifier assigned by registration agencies, such as ANSI or ISO. For an explanation of the standard identifiers, see the current LDAP standards available through the IETF Web site.
Description	This optional field is for your information only.
Syntax	Type the standardized rules for data entry applicable to this attribute type.
Size	Type the maximum size allowed for this object.
Single Value	Select this check box to indicate that this attribute type contains a maximum of one value.

4. Select the Advanced tab. Enter values in each of the fields as described in the following table.

Field	Description
Indexed	Select to add this attribute to the index, thereby making it available for use in a search. Only those attributes that have an equality matching rule can be indexed.

Field	Description
Usage	Specify standards for how the attribute can be used. Options are: ■ `userApplications` Attributes whose values must be entered by the user, for example, `telephoneNumber` ■ `directoryOperation` Attributes whose values are entered by the directory server, for example, `creatorName` or `timeStamp` ■ `distributedOperation` ■ `dSAOperation` Attributes used for the internal operation of the server, for example, orclUpdateSchedule
Ordering	Specify standards for how precedence is established for values
Equality	Specify standards for how equality is determined in compare and search operations
Substring	Specify regular expression matching
Super	Add the super attribute for this attribute. To do this: **1.** Click the Add button next to this field. The Super Attribute Selector appears. **2.** Select the super attribute and click Select. **3.** Repeat as needed. To delete a super attribute from the Super field, select it, then click Delete.

5. Click OK.

> **Note:** To use this attribute, remember to declare it to be part of the attribute set for an object class. You do this by selecting Schema Management in the navigator pane, then, in the right pane, selecting the Object Classes tab page. For further instructions, see "Guidelines for Modifying Object Classes" on page 7-4.

Creating a New Attribute from an Existing One by Using Oracle Directory Manager

To add an attribute by copying an existing attribute:

1. In the navigator pane, select Schema Management.
2. In the right pane, select the Attributes tab.
3. In the Attributes tab page, select the attribute you want to copy.
4. Click the Create Like button at the bottom of the right pane. The New Attribute Type dialog box for that attribute appears. This dialog box contains two tab pages—General and Advanced—with fields in which you enter values either by typing or selecting from menus.
5. Select the General tab and enter values in each of the fields as described in the following table. You must always change the DN to that of the new attribute.

Field	Description
Name	Type the name for this attribute.
Object ID	Type the Object ID for this attribute. The Object ID is a standardized numerical sequence based on IETF standards. It must be unique. Normally this is derived from the identifier assigned by registration agencies, such as ANSI or ISO. For an explanation of the standard identifiers, see the current LDAP standards available through the IETF Web site.
Description	This optional field is for your information only.
Syntax	Type the standardized rules for data entry applicable to this attribute type.
Size	Type the maximum size allowed for this object.
Single Value	Select this check box to indicate that this attribute type contains a maximum of one value.

6. Select the Advanced tab and enter values in each of the fields as described in the following table.

Field	Description
Indexed	Select to add this attribute to the index, thereby making it available for use in a search. Only those attributes that have an equality matching rule can be indexed.
Usage	Specify standards for how the attribute can be used. Options are: ■ `userApplications` Attributes whose values must be entered by the user, for example, `telephoneNumber` ■ `directoryOperation` Attributes whose values are entered by the directory server, for example, `creatorName` or `timeStamp` ■ `distributedOperation` ■ `dSAOperation` Attributes used for the internal operation of the server, for example, orclUpdateSchedule
Ordering	Specify standards for how precedence is established for values
Equality	Specify standards for how equality is determined in compare and search operations
Substring	Specify regular expression matching
Super	Add the super attribute for this attribute. To do this: 1. Click the Add button next to this field. The Super Attribute Selector appears. 2. Select the super attribute and click Select. 3. Repeat as needed. To delete a super attribute from the Super field, select it, then click Delete.

7. Click OK.

Modifying an Attribute by Using Oracle Directory Manager

To modify an attribute by using Oracle Directory Manager:

1. In the navigator pane, select Schema Management.
2. In the right pane, select the Attributes tab, then select an editable attribute in the list.
3. Click Edit. The Attribute dialog box displays two tab pages—General and Advanced—with fields in which you enter values either by typing or selecting from menus.
4. Select the General tab and enter values in each of the fields as described in the following table.

Field	Description
Name	Type the name for this attribute.
Object ID	Type the Object ID for this attribute. The Object ID is a standardized numerical sequence based on IETF standards. It must be unique. Normally this is derived from the identifier assigned by registration agencies, such as ANSI or ISO. For an explanation of the standard identifiers, see the current LDAP standards available through the IETF Web site.
Description	This optional field is for your information only.
Syntax	Type the standardized rules for data entry applicable to this attribute type.
Size	Type the maximum size allowed for this object.
Single Value	Select this check box to indicate that this attribute type contains a maximum of one value.

5. Select the Advanced tab and enter values in each of the fields as described in the following table.

Field	Description
Indexed	Select to add this attribute to the index, thereby making it available for use in a search. Only those attributes that have an equality matching rule can be indexed.
Usage	Specify standards for how the attribute can be used. Options are: ■ `userApplications` Attributes whose values must be entered by the user, for example, `telephoneNumber` ■ `directoryOperation` Attributes whose values are entered by the directory server, for example, `creatorName` or `timeStamp` ■ `distributedOperation` ■ `dSAOperation` Attributes used for the internal operation of the server, for example, orclUpdateSchedule
Ordering	Specify standards for how precedence is established for values
Equality	Specify standards for how equality is determined in compare and search operations
Substring	Specify regular expression matching
Super	Add the super attribute for this attribute. To do this: 1. Click the Add button next to this field. The Super Attribute Selector appears. 2. Select the super attribute and click Select. 3. Repeat as needed. To delete a super attribute from the Super field, select it, then click Delete.

6. Click OK.

Deleting an Attribute by Using Oracle Directory Manager

To delete an attribute:

1. In the navigator pane, select Schema Management.
2. In the right pane, select the Attributes tab, then select an editable attribute in the list.
3. Click Delete.

Indexing an Attribute by Using Oracle Directory Manager

Oracle Internet Directory uses indexes to make attributes available for searches. When Oracle Internet Directory is installed, certain attributes are already indexed. If you want to use additional attributes in search filters, you must index them.

Note: You can use Oracle Directory Manager to index an attribute only at the time when you create it. You cannot use Oracle Directory Manager to index an already existing attribute. To index an already existing attribute, use the Catalog Management tool.

Also, you can index only those attributes that have an equality matching rule.

See Also: "Indexing an Attribute by Using Command-Line Tools" on page 7-30 for instructions on using the command-line catalog management tool

Viewing Indexed Attributes by Using Oracle Directory Manager

To view indexed attributes:

1. In the navigator pane, select Schema Management.
2. In the right pane, select the Attributes tab. The Attributes tab displays all of the attributes in the schema. A selected check box in the Indexed column indicates an indexed attribute.

Adding an Index to an Attribute by Using Oracle Directory Manager

When you create an attribute as described in "Adding an Attribute by Using Oracle Directory Manager" on page 7-21, you use the New Attribute Type dialog box. On the Advanced tab page of that dialog box, you select the Indexed check box.

Dropping an Index from an Attribute by Using Oracle Directory Manager

To drop an index from an attribute:

1. In the navigator pane, select Schema Management.
2. In the right pane, select the Attributes tab.
3. Select the indexed attribute. Note that this must be an attribute that is editable as indicated by the icon to the left of the attribute name.
4. Click Drop Index.

Managing Attributes by Using Command-Line Tools

This section discusses adding, modifying, and indexing attributes by using command-line tools. This section contains these topics:

- Adding and Modifying Attributes by Using ldapmodify
- Indexing an Attribute by Using Command-Line Tools

Adding and Modifying Attributes by Using ldapmodify

> **See Also:** "ldapmodify Syntax" on page A-15 for a detailed explanation of this command and its options

To add a new attribute to the schema by using ldapmodify, type a command similar to the following at the system prompt:

In this example, the LDIF file, `my_ldif_file.ldi`, contains data similar to this:

```
dn: cn=subschemasubentry
changetype: modify
add: attributetypes
attributetypes: ( 1.2.3.4.5 NAME 'myattr' SYNTAX
                  '1.3.6.1.4.1.1466.115.121.1.38' )
```

To specify an attribute as single-valued, include in the attribute definition entry in the LDIF file the keyword SINGLE-VALUE with surrounding white space.

To load the file, you would enter this command:

```
ldapmodify -h my_host -p my_port_number -f my_ldif_file.ldi
```

You can find a given syntax Object ID by using either Oracle Directory Manager or the ldapsearch command-line tool.

> **See Also:** In the right pane, select the Matching Rules tab. The fields in this tab page are shown as column heads. They are: on page 7-31 for instructions on how to view syntaxes by using either Oracle Directory Manager or ldapsearch

Indexing an Attribute by Using Command-Line Tools

Oracle Internet Directory uses indexes to make attributes available for searches. When Oracle Internet Directory is installed, the entry `cn=catalogs` lists available attributes that can be used in a search.

If you want to use additional attributes in search filters, you must add them to the catalog entry. Only those attributes that have an equality matching rule can be indexed.

You can index a new attribute—that is, one for which no data exists in the directory—by using ldapmodify. You can index an attribute for which data already exists in the directory by using the Catalog Management tool. You can drop an index from an attribute by using ldapmodify, but Oracle Corporation recommends that you use the Catalog Management tool.

Indexing an Attribute for Which *No* Data Exists by Using ldapmodify

Once you have defined a new attribute in the schema, you can add it to the catalog entry by using ldapmodify.

To add an attribute for which no directory data exists by using ldapmodify, import an LDIF file by using ldapmodify. For example, to add a new attribute `foo` that has already been defined in the schema, import the following LDIF file by using ldapmodify:

```
dn: cn=catalogs
changetype: modify
add: orclindexedattribute
orclindexedattribute: foo
```

You should not use this method to index an attribute for which data exists in the directory. To index such an attribute, use the Catalog Management Tool.

To drop an index from an attribute by using ldapmodify, specify `delete` in the LDIF file. For example:

```
dn: cn=catalogs
changetype: modify
delete: orclindexedattribute
orclindexedattribute: foo
```

See Also: "ldapmodify Syntax" on page A-15

Indexing an Attribute for Which Data Exists by Using the Catalog Management Tool

Use the Catalog Management Tool to index an attribute for which data already exists and to drop an index from an attribute.

See: "Catalog Management Tool Syntax" on page A-32

Viewing Matching Rules

This section contains these topics:

- Viewing Matching Rules by Using Oracle Directory Manager
- Viewing Matching Rules by Using ldapsearch

Note: Matching rules cannot be modified.

Viewing Matching Rules by Using Oracle Directory Manager

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server instance*, then select Schema Management.
2. In the right pane, select the Matching Rules tab. The fields in this tab page are shown as column heads. They are:

Name	Name of the attribute matching rule
Object ID	Unique identifier of this matching rule
Description	Words describing the matching rule (optional)
Syntax	Syntax used with this matching rule

Viewing Matching Rules by Using ldapsearch

Use ldapsearch on the subentry `cn=subSchemaSubentry`.

See Also: "ldapsearch Syntax" on page A-22

Viewing Syntaxes

This section contains these topics:

- Viewing Syntaxes by Using Oracle Directory Manager
- Viewing Syntaxes by Using by Using ldapsearch

Note: Syntaxes cannot be modified.

Viewing Syntaxes by Using Oracle Directory Manager

To view syntaxes by using Oracle Directory Manager:

1. In the navigator pane, select Schema Management.
2. In the right pane, select the Syntaxes tab. The fields in this tab page are shown as column heads. They are:
   - Description—Name of the attribute syntax
   - Object ID—Unique identifier of this syntax

Viewing Syntaxes by Using by Using ldapsearch

Use ldapsearch on the subentry `cn=subSchemaSubentry`.

See Also: "ldapsearch Syntax" on page A-22

8

Managing Directory Entries

This chapter explains how to view, add, modify, and delete entries.

This chapter contains these topics:

- Managing Entries by Using Oracle Directory Manager
- Managing Entries by Using Command-Line Tools
- Managing Entries by Using Bulk Tools
- Managing Knowledge References (Referrals)

> **See Also:** Chapter 2, "Concepts and Architecture" for an overview of directory entries, directory information trees, distinguished names, and relative distinguished names

Managing Entries by Using Oracle Directory Manager

This section contains these topics:

- Searching for Entries by Using Oracle Directory Manager
- Searching for Audit Log Entries by Using Oracle Directory Manager
- Viewing Attributes for a Specific Entry by Using Oracle Directory Manager
- Adding Entries by Using Oracle Directory Manager
- Modifying Entries by Using Oracle Directory Manager
- Managing Entries with Attribute Options by Using Oracle Directory Manager

Searching for Entries by Using Oracle Directory Manager

You can display all entries by using the navigator pane, or search for one or more specific entries by using the Oracle Directory Manager search feature.

To display an entry, in the navigator pane, expand Oracle Internet Directory Servers > *directory server instance* > Entry Management to display its subtree.

The root of the tree is listed first, then the second level, and so forth, moving from left to right. The subtree lists the **RDN** of each entry in hierarchical order. To see the lower level entries within any subtree, click the plus sign (+) to the left of the parent entry.

To search for a directory entry:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server instance*, and select Entry Management. The Search fields appear in the right pane.
2. In the Root of the Search field, enter the **DN** of the root of your search.

   For example, suppose you want to search for an employee who works in the Manufacturing division in the IMC organization in the Americas. The DN of the root of your search would be:

   ```
   ou=Manufacturing,ou=Americas,o=IMC,c=US
   ```

   You would therefore type that DN in the Root of the Search text box.

You can also select the root of your search by browsing the **directory information tree (DIT)**. To do this:

a. Click Browse to the right of the Root of the Search field. The Select Distinguished Name (DN) Path: Tree View dialog box appears.

b. Click the plus sign (+) next to tree view to display its entries.

c. Continue navigating to the entry that represents the level you want for the root of your search.

d. Select that entry, then click OK. The DN for the root of your search appears in the Root of the Search text box in the right pane.

3. In the Max Results (entries) box, type the maximum number of entries you want your search to retrieve. The default is 200. The directory server retrieves the value you set, up to 1000.

4. In the Max Search Time (seconds) box, type the maximum number of seconds for the duration of your search. The value you enter here must be at least that of the default, namely, 25. The directory server searches for the amount of time you specify, up to one hour.

5. In the Search Depth list, select the level in the DIT to which you want to search.

   The options are:

   - Base: Retrieves a particular directory entry. Along with this search depth, you use the Search criteria bar to select the attribute `objectClass` and the filter `Present`.
   - One Level: Limits your search to all entries beginning one level down from the root of your search
   - Subtree: Searches entries within the entire subtree, including the root of your search

6. In the Search Criteria box, use the lists and text fields on the search criteria bar to focus your search.

a. From the list at the left end of the search criteria bar, select an attribute of the entry for which you want to search. Because not all attributes are used in every entry, be sure that the attribute you specify actually corresponds to one in the entry for which you are looking. Otherwise, the search will fail.

b. From the list in the middle of the search criteria bar, select a filter. Options are:

Filter	Description
Begins With	Searches by using only the first few characters of the attribute's value. For example, `cn Begins With Fran` retrieves all entries in which the first few letters of the `cn` attribute are `Fran`. These would include such names as Frank, Fran, Frances, and Franklin.
Ends With	Searches for an entry by using only the last few characters of the specified attribute's value. For example, `cn Ends With son` retrieves Baldisson, Jacobson, and Johnson.
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter. For example, `cn Contains Wins` retrieves all entries in which the `cn` attribute contains the letters `wins`. These would include Winslow, Czerwinski, and Winship.
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter. For example, `cn Exactly Matches Franklin Baldwins` retrieves all entries in which the `cn` attribute has the value `Franklin Baldwins`.
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. For example, `cn Greater or Equal Frank` retrieves all entries with `cn` attributes that range from the first Frank to the end of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. For example, `cn Less or Equal Frank` retrieves all `cn` attributes from the first Frank to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. The phrase `cn Present` retrieves all entries with the `cn` attribute at that level of the tree.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you just selected. For example, if the attribute you selected was `cn`, you could type the particular common name you want to find.

7. To further refine your search, use the buttons in the Search Criteria box to enhance the search criteria bar.

Button	Description
New	Creates a new search criteria bar in the Search Criteria field. This button is enabled only when the Search Criteria field is empty.
And	Creates another search criteria bar in the Search Criteria field. Matches all entries with one specified attribute with those that also have another specified attribute. For example, `cn=Baldwins And title=Laborer` retrieves all Baldwins who are also laborers.
Or	Creates another search criteria bar in the Search Criteria field. Matches all entries with either one specified attribute or another. For example, `title=Laborer Or title=Foreman` retrieves all employees who are either laborers or foremen.
Not	Negates the criterion in the selected search criteria bar and retrieves all entries that do not have the specified criterion. For example, `cn=Frank And Not title=Laborer` retrieves all persons named Frank who are not laborers.
Delete	Deletes a selected search criteria bar
Advanced	Adds a search criteria bar when including attribute options in the search. Use this syntax: *`attribute`*;*`attribute_option filter attribute_option_value`* For example, cn;lang_sp=J* retrieves all attribute option values for `cn;lang_sp=` that begin with the letter J. **Note:** Before an attribute option can be used in searches, the parent attribute of that attribute option must be indexed. For example, in the case of the attribute option `carLicense;lang_sp`, the `carLicense` attribute must be indexed before the `carLicense;lang_sp` attribute option can be used in searches. **See Also:** ■ "Indexing an Attribute by Using Oracle Directory Manager" on page 7-28 ■ "Indexing an Attribute by Using Command-Line Tools" on page 7-30

8. Click Search. The results of your search appear in the Distinguished Name box.

> **See Also:** "Configuring Searches" on page 6-25 for instructions on setting the number of entries to display in searches, and to set the time limit for searches

Searching for Audit Log Entries by Using Oracle Directory Manager

You can also search for audit log entries by using Oracle Directory Manager.

To use Oracle Directory Manager to view audit log entries:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*, and select Audit Log Management. The corresponding right pane appears.
2. In the Max Results (entries) field, type the maximum number of entries you want your search to retrieve. The default is 200. The directory server retrieves the number you specify, up to 1000.
3. In the Max Search Time (seconds) box, type the maximum number of seconds for the duration of your search. The value you enter here must be at least that of the default, namely, 25. The directory server searches for the amount of time you specify, up to one hour.
4. In the Search Criteria box, use the lists and text fields on the search criteria bar to focus your search.
   a. From the list at the left end of the search criteria bar, select an attribute of the entry for which you want to search. Because not all attributes are used in every entry, be sure that the attribute you specify actually corresponds to one in the entry for which you are searching. Otherwise, the search fails.
   b. From the list in the middle of the search criteria bar, select a filter. Options are:

Filter	Description
Begins With	Searches by using only the first few characters of the attribute's value.
Ends With	Searches for an entry by using only the last few characters of the specified attribute's value.
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter.
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter.
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter.

Filter	Description
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you just selected. For example, if the attribute you selected was `cn`, you could type the particular common name you want to find.

5. To further refine your search, use the buttons in the Search Criteria box to enhance the search criteria bar.

Button	Description
New	Creates a new search criteria bar in the Search Criteria field. This button is enabled only when the Search Criteria field is empty.
And	Creates another search criteria bar in the Search Criteria field. Matches all entries with one specified attribute with those that also have another specified attribute. For example, `cn=Baldwins And title=Laborer` retrieves all Baldwins who are also laborers.
Or	Creates another search criteria bar in the Search Criteria field. Matches all entries with either one specified attribute or another. For example, `title=Laborer Or title=Foreman` retrieves all employees who are either laborers or foremen.
Not	Negates the criterion in the selected search criteria bar and retrieves all entries that do not have the specified criterion. For example, `cn=Frank And Not title=Laborer` retrieves all persons named Frank who are not laborers.
Delete	Deletes a selected search criteria bar

6. Click Search. The results of your search appear in the Distinguished Name box.

7. To view the properties of a particular audit log entry, select it in the Distinguished Name box, then click View Properties. The Audit Log Entry dialog box displays the properties for the audit log entry you selected.

> **See Also:** "Configuring Searches" on page 6-25 for instructions on setting the number of entries to display in searches, and to set the time limit for searches

Viewing Attributes for a Specific Entry by Using Oracle Directory Manager

Once you have displayed the results of your search, click the entry whose attributes you want to view. An Entry dialog box displays the attributes for that entry.

Some attributes can also be DNs. For example, one attribute for a given employee might be that employee’s manager who, in turn, has a DN. In this case, when you display the Entry dialog box for the employee, you would see a Browse button next to the Manager text box. To find information about that manager, click Browse to display the Directory: Entry Management dialog box, then follow the steps mentioned in "Searching for Entries by Using Oracle Directory Manager" on page 8-2.

> **See Also:** "Viewing All Directory Attributes by Using Oracle Directory Manager" on page 7-18 for instructions about how to view all attributes in the directory

Adding Entries by Using Oracle Directory Manager

Adding a New Entry by Using Oracle Directory Manager

To add or delete entries with Oracle Directory Manager, you must have write access to the parent entry and you must know the DN for the new entry.

To add a new entry:

1. Expand Oracle Internet Directory Servers > *directory_server_instance*, then select Entry Management.
2. On the toolbar, click Create. The New Entry dialog box appears.
3. In the Distinguished Name field, type the full DN. You may also click Browse to locate and select the DN of the parent for the entry you want to add. The entry you select appears in the Distinguished Name field. To the left of that parent DN, type the RDN for your new entry, followed by a comma.
4. To specify the **object classes** for the new entry, next to the Object Classes box, click Add. The Super Class Selector dialog box appears.
5. In the Super Class Selector dialog box, select an object class, then click Select. As you select from the object class list, mandatory and optional attributes populate the windows in the tab pages in the lower half of the New Entry dialog box. You must enter values into the mandatory attributes fields. You are not required to enter values into the optional attributes fields.

6. When you have selected the object classes and provided values for the appropriate attributes, click OK.

Adding an Entry by Copying an Existing Entry in Oracle Directory Manager

You can use Oracle Directory Manager to create a new entry by copying from an existing entry and changing its DN. When you do this, you should also change the attributes, such as name and address, so that they correspond to the new DN. To add an entry, you must have write access to its parent.

> **Tip:** You can find a template for the new DN by looking up other similar entries in the search pane.

To add an entry by copying an existing entry:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*, then select Entry Management. in the right pane, the Search interface appears. Use it to search for an entry that you want to use as a template.
2. From the entries retrieved, double-click one that you want to use as your template. The Entry dialog box for that entry appears.
3. In the Entry dialog box, click Create Like. A New Entry: Create Like dialog box appears.
4. Change critical fields to tailor this entry to the one that you want to create. You must always change the DN and the common name in this operation, or the pane will not save your new entry data. For example, if you create an entry for Henri Latrobe by using the entry for Henri Latour as the template, then you have to change `cn=Henri Latour` in the DN to `cn=Henri Latrobe`. You also must change any other attributes that must be unique, such as employee number and telephone number.
5. Click OK to save your changes.

> **See Also:** The online help for this dialog box for details about adding information into fields

Example: Adding a User Entry by Using Oracle Directory Manager

In this example, we create a user named Anne Smith and assign her a password.

1. Login as the administrator.
2. Expand Oracle Internet Directory Services > directory*_server_instance*, and select Entry Management.
3. On the toolbar, click the Create button. The New Entry dialog box appears.
4. In the Distinguished Name field, type the full DN. You may also click the Browse button to locate the DN of the parent for this entry, then type the RDN, namely, `cn=Anne Smith`, followed by a comma, to the left of that parent DN.
5. To the right of the Object Classes box, Click Add. The Super Class Selector dialog box appears.
6. In the Super Class Selector dialog box, select the `person` object class, then click Select. This returns you to the New Entry dialog box.
7. In the New Entry dialog box, click the Optional Properties tab, and scroll to the `userPassword` window.
8. Type the password for Anne Smith.

Adding Group Entries by Using Oracle Directory Manager

A group entry is one that contains a list of entries, for example, an e-mail list. You associate it with either the `groupOfNames` or `groupOfUniqueNames` object class, which has the object class `orclPrivilegeGroup` as a subclass.

You determine membership in the group by adding DNs to the multivalued attribute `member` if the entry belongs to the `groupOfNames` object class, or `uniqueMember` if the entry belongs to the `groupOfUniqueNames` object class.

To add a group entry:

1. Expand Oracle Internet Directory Servers > *directory_server_instance*, then select Entry Management.
2. On the toolbar, click Create. The New Entry dialog box appears.
3. In the Distinguished Name field, type the full DN. You may also use the Browse button to locate the DN of the parent for the entry you want to add, then type the RDN for the new entry, followed by a comma, to the left of that parent DN.
4. To specify the object classes you want to use for the new entry, to the right of the Object Classes box, click Add. The Super Class Selector dialog box appears.

5. In the Super Class Selector dialog box, select the `top` object class, then click the Select button. The `top` object class appears in the Object Classes box of the New Entry dialog box.
6. In the same way:
   a. To the right of the Object Classes box, click Add.
   b. From the Super Class Selector dialog box, select the `groupOfNames` or `groupOfUniqueNames` object class.
   c. Click Select. The object class you selected appears in the Object Classes window of the New Entry dialog box.
7. Enter the mandatory and optional attributes for your group entry.

   If you selected the `groupOfNames` object class, a Browse button appears next to some of the fields, for example, the member field on the Mandatory Properties tab page. To enter a mandatory property by browsing:

   a. Click Browse. The Directory: Entry Management dialog box appears.
   b. Use this dialog box to search for a particular entry you want to add to the list.
   c. In the Distinguished Name window of the Directory: Entry Management dialog box, select the entry, then click OK. This returns you to the New Entry dialog box. The entry you just selected is added to the list in the members window.
8. Click OK.

See Also:

- "Searching for Entries by Using Oracle Directory Manager" on page 8-2 for instructions on using the search pane
- "Privilege Groups" on page 13-3 for instructions on setting access control policies for group entries
- Globalization Support on page 2-14 and Chapter 13, "Managing Directory Access Control" for information about access privileges

Modifying Entries by Using Oracle Directory Manager

Oracle Directory Manager is governed by standard LDAP conventions, including the following:

- Once you have assigned object classes to an entry and populated its attributes with data, you cannot change those object classes that are used by that entry.

  For example, if you configure an entry to use object classes `Person` and `Organizational Role`, you cannot later add another object class to this entry.

- You cannot add mandatory attributes to an object class already in use by some entries. You may add optional attributes to object classes that are already in use by entries. If you add optional attributes to an object class already in use by some entries, no special rules apply—they are added as empty attributes to those entries.

To modify an entry:

1. Perform a search for the entry you want to modify as described in "Searching for Entries by Using Oracle Directory Manager" on page 8-2.
2. In the Distinguished Name box of the right pane, select the entry you want to modify.
3. Click Edit. The Entry dialog box appears.
4. Select the Properties tab page. If you do not see the attributes you want to add or modify, then, at the top of the tab page, select View Properties: All.
5. In the Properties tab page, modify the values of any editable attributes.
6. Click OK.

Example: Modifying a User Entry by Using Oracle Directory Manager

In this example, we modify the password for the entry we created for Anne Smith in the section "Example: Adding a User Entry by Using Oracle Directory Manager" on page 8-10.

1. Perform a search for the Anne Smith entry.
2. In the right pane, in the Distinguished Name box, select the entry for Anne Smith.
3. Click Edit.

4. In the Entry dialog box, scroll to the `userPassword` window and modify the value.
5. Click OK.

Managing Entries with Attribute Options by Using Oracle Directory Manager

This section tells you how to add, modify, and delete attribute options.

> **See Also:** "Searching for Entries by Using Oracle Directory Manager" on page 8-2 for instructions on searching for entries with attribute options

Adding an Attribute Option to an Existing Entry by Using Oracle Directory Manager

> **Note:** In Oracle Internet Directory release 3.0.1, Oracle Directory Manager does not allow you to add an attribute option to an entry when you create the entry. You can use Oracle Directory Manager to add attribute options only to already existing entries.

To add an attribute option to an existing entry:

1. Expand Oracle Internet Directory Servers > *directory server instance* > Entry Management, then select the entry to which you want to add an attribute option. The corresponding tab pages appear in the right pane.
2. In the right pane, in the Properties tab page, in the View Properties field, select Advanced. The Properties tab page changes accordingly.
3. In the Attribute field, select the attribute to which you want to add the option, for example, `ou`.
4. In the Attribute Options field, enter the attribute option, for example, `lang-en`.
5. In the Attribute Value field, enter the value of the attribute option you just specified, for example, `Server Technologies`. To add more than one attribute value for the specified attribute option, separate the values by using a semicolon.
6. Click Apply.

Modifying an Attribute Option by Using Oracle Directory Manager

To modify an attribute option:

1. Expand Oracle Internet Directory Servers > *directory server instance* > Entry Management, then select the entry from which you want to delete an attribute option. The corresponding tab pages appear in the right pane.
2. In the Properties tab page, in the View Properties field, select either Only Non-null Values or All.
3. Scroll to the field containing the attribute option you want to modify.
4. Modify the value in the field.
5. Click Apply.

Deleting an Attribute Option by Using Oracle Directory Manager

To delete an attribute option:

1. Expand Oracle Internet Directory Servers > *directory server instance* > Entry Management, then select the entry from which you want to delete an attribute option. The corresponding tab pages appear in the right pane.
2. In the Properties tab page, in the View Properties field, select either Only Non-null Values or All.
3. Scroll to the field containing the attribute option you want to delete.
4. Delete the value in the field.
5. Click Apply.

Managing Entries by Using Command-Line Tools

This section points you to the command-line tools you can use in managing entries. It also provides several examples of entry management by using command-line tools. It contains these topics:

- Command-Line Tools for Managing Entries
- Example: Adding a User Entry by Using ldapadd
- Example: Adding an Attribute Option by Using ldapmodify
- Example: Modifying a User Entry by Using ldapmodify
- Managing Entries with Attribute Options by Using Command-Line Tools

Command-Line Tools for Managing Entries

The following table lists each of the command-line tools, and tells you where to find syntax and usage notes for each one.

Tool	Task(s)	Syntax and Usage Notes
ldapsearch	Search for directory entries.	"ldapsearch Syntax" on page A-22
ldapbind	Authenticate a user or client to a directory server. Verify that you can connect a client to a server.	"ldapbind Syntax" on page A-8
ldapadd	Add entries one at a time. Add new configuration set entries. Configure a server with an input file.	"ldapadd Syntax" on page A-4
ldapaddmt	Add several entries concurrently by using this multithreaded tool.	"ldapaddmt Syntax" on page A-6
ldapmodify	Create, update, and delete attribute data for an entry. Modify configuration set entries. Modify DN or RDN of an entry.	"ldapmodify Syntax" on page A-15
ldapmodifymt	Modify several entries concurrently by using this multithreaded tool.	"ldapmodifymt Syntax" on page A-20
ldapdelete	Delete entries.	"ldapdelete Syntax" on page A-11
ldapcompare	Compare attribute values you specify with those in a directory entry.	"ldapcompare Syntax" on page A-9

Tool	Task(s)	Syntax and Usage Notes
ldapmoddn	Modify the DN or RDN of an entry. Rename an entry or a subtree. Move an entry or a subtree under a new parent.	"ldapmoddn Syntax" on page A-13

Example: Adding a User Entry by Using ldapadd

The following example shows an LDIF file, named entry.ldif, for the user entry for an employee named John:

```
dn: cn=john, c=us
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: john
cn;lang-fr:Jean
cn;lang-en-us:John
sn: Doe
jpegPhoto: /photo/john.jpg
userpassword: welcome
```

This file contains the cn, sn, jpegPhoto, and userpassword attributes.

For the cn attribute, it specifies two options: cn;lang-fr, and cn;lang-en-us. These options return the common name in either French or American English.

For the jpegPhoto attribute, it specifies the path and file name of the corresponding JPEG image you want to include as an entry attribute.

Example: Modifying a User Entry by Using ldapmodify

The following example changes the password for a user named Audrey from welcome to audreyspassword. As in the example above, the data for this user entry is in the entry.ldif file. This file contains the following:

```
dn: cn=audrey,c=us
changetype: modify
replace: userpassword
userpassword: audreyspassword
```

Issue this command to modify the file:

```
ldapmodify -h myhost -p 389 -b -f entry.ldif
```

Managing Entries with Attribute Options by Using Command-Line Tools

This section provides examples of how to add and delete attribute options, and how to search for entries with attribute options.

Example: Adding an Attribute Option by Using ldapmodify

Suppose that you were adding the Spanish equivalent of an entry for John, and that the data for this user entry is in the `entry.ldif` file. This file contains the following:

```
dn: cn=john,c=us
changeType: modify
add: cn;lang-sp
cn;lang-sp: Juan
```

Issue this command to modify the file:

```
ldapmodify -h myhost -p 389 -b  -f entry.ldif
```

Example: Deleting an Attribute Option by Using ldapmodify

The following example deletes the `cn;lang-fr` attribute option from the entry for John. As in the previous example, assume that the data for this user entry is in the `entry.ldif` file. This file contains the following:

```
dn: cn=john, c=us
changetype: modify
delete: cn;lang-fr
cn;lang-fr: Jean
```

Issue this command to modify the file:

```
ldapmodify -h myhost -p 389 -b  -f entry.ldif
```

Example: Searching for Entries with Attribute Options by Using ldapsearch

The following example retrieves entries with common name (`cn`) attributes that have an option specifying a language code attribute option. This particular example retrieves entries in which the common names are in French and begin with the letter R.

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub "cn;lang-fr=R*"
```

Suppose that, in the entry for John, no value is set for the `cn;lang-it` language code attribute option. In this case, the following example fails:

```
ldapsearch -p 389 -h myhost -b "c=us" -s sub "cn;lang-it=Giovanni
```

See Also: "Attribute Options" on page 2-7

Managing Entries by Using Bulk Tools

This section lists and describes some of the more common tasks you perform with bulk tools.

This section contains these topics:

- Importing an LDIF File by Using bulkload
- Converting Directory Data to LDIF
- Modifying a Large Number of Entries
- Deleting a Large Number of Entries

See Also: "Using Bulk Tools" on page 5-13 for an overview of these tools

Importing an LDIF File by Using bulkload

To import an LDIF file, you use the bulkload utility. This section discusses the tasks to process an LDIF file through bulkload.

Note: The bulkload utility expects an empty directory and will either fail or overwrite if there are existing entries.

Before performing a bulk load, stop the Oracle Internet Directory processes. See Chapter 4, "Preliminary Tasks" for instructions on stopping directory server instances.

This section contains these topics:

- Task 1: Back Up the Oracle Server
- Task 2: Find Out the Oracle Internet Directory Password
- Task 3: Check Input for Schema and Data Consistency Violations
- Task 4: Generate the Input Files for SQL*Loader

- Task 5: Load the Input Files
- If Bulk Loading Fails

Task 1: Back Up the Oracle Server

Before you import the file, back up the Oracle database server as a safety precaution.

> **See Also:** *Oracle9i User-Managed Backup and Recovery Guide*

Task 2: Find Out the Oracle Internet Directory Password

To use bulkload and the other shell script tools that have commands that end with .sh, you must provide the Oracle Internet Directory password. The default password is ods, although the system administrator can change it by using the **OID Database Password Utility**.

> **See Also:** "Using the OID Database Password Utility" on page 5-14

Task 3: Check Input for Schema and Data Consistency Violations

On Solaris, the bulkload.sh file usually resides in $*ORACLE_HOME*/ldap/bin. On Windows NT, this file usually resides in *ORACLE_HOME*\ldap\bin.

Check the input file by typing:

```
bulkload.sh -connect net_service_name -check path_to_ldif-filename
```

All schema violations are reported in $*ORACLE_HOME*/ldap/log/schemacheck.log

If any violations are detected in the input file, use an ASCII text file editor to fix or remove them. If there are any duplicate entries, their DNs are logged in $*ORACLE_HOME*/ldap/log/duplicate.log.

Task 4: Generate the Input Files for SQL*Loader

After you have fixed any errors in the input file, rerun bulkload with the -generate option as shown in the following example. During this step, LDIF data is converted to SQL*Loader specific format.

```
bulkload.sh -connect net_service_name -generate ldif-filename
```

All loading errors are reported in
`$ORACLE_HOME/ldap/log`

When this command completes successfully, it generates `*.dat` files in the `$ORACLE_HOME/ldap/load` directory to be used by SQL*Loader in `-load` mode. Do not modify these files.

Task 5: Load the Input Files

After you have generated the input files, rerun bulkload with the `-load` option. During this step, the `*.dat` files, which are in Oracle SQL*Loader specific format, are loaded into the database and the attribute indexes are created. The syntax is:

```
bulkload.sh -connect net_service_name -load
```

If Bulk Loading Fails

All loading errors are reported in the `$ORACLE_HOME/ldap/log/directory` with the file extension `.bad`.
If bulk loading fails, the database could be left in an inconsistent state. It may be necessary to restore the database to its state prior to the bulk loading operation.

Converting Directory Data to LDIF

Converting directory data to LDIF by using LDIF Writer makes the data available for loading into a new node in a replicated directory or into another node for backup storage.

See Also: "ldifwrite Syntax" on page A-31

Modifying a Large Number of Entries

The bulkmodify utility enables you to modify a large number of existing entries efficiently.

See Also: "bulkmodify Syntax" on page A-29

Deleting a Large Number of Entries

The bulkdelete utility enables you to delete an entire subtree efficiently.

See Also: "bulkdelete Syntax" on page A-27

Managing Knowledge References (Referrals)

A **knowledge reference**, also called a **referral**, is represented in the directory as a particular type of **entry**. When you create a knowledge reference entry, you associate it with the `referral` and `extensibleObject` **object classes**. Typically, you create knowledge reference entries at the place in the **DIT** where you want to establish the partition.

Knowledge references provide users with LDAP URLs. You enter these URLs as values for the `ref` attribute. There can be multiple `ref` attributes specified for any knowledge reference entry. Similarly, there can be multiple knowledge reference entries in the DIT.

> **See Also:** "Partitioning" on page 2-25 for an overview of knowledge references and a description of **smart knowledge references** and **default knowledge references**

This section contains these topics:

- Configuring Smart Knowledge References
- Configuring Default Knowledge References

Configuring Smart Knowledge References

A search result can contain regular entries along with knowledge references. When a user performs a search operation, Oracle Internet Directory looks for the knowledge reference entry within the specified scope of the search. If it finds the knowledge reference, then Oracle Internet Directory returns it to the client.

If a user performs an add, delete, or modify operation on an entry located below the knowledge reference entry, then Oracle Internet Directory returns the knowledge reference.

For example, suppose you want to partition the DIT based on the geographical location of the directory servers. In this example, assume that:

- The `c=us` naming context is held locally on Server A and Server B in the United States.
- The `c=uk` naming context is held locally on Server C and Server D in the United Kingdom.

In this case, you would configure knowledge references between these two naming contexts as follows:

1. On Server A in the United States, configure a knowledge reference for the c=uk object on Server C and Server D:

```
dn: c=uk
c: uk
ref: ldap://host_C:389/c=uk
ref: ldap://host_D:686/c=uk
objectclass: top
objectclass: referral
objectClass: extensibleObject
```

2. Configure a similar knowledge reference on Server C in the United Kingdom for the c=us object on Server A and Server B:

```
dn: c=us
c: us
ref: ldap://host_A:4000/c=us
ref: ldap://host_B:5000/c=us
objectclass: top
objectclass: referral
objectClass: extensibleObject
```

Results:

- A client querying Server A with base o=foo,c=uk receives a knowledge reference
- A client querying Server C with base o=foo,c=us receives a knowledge reference
- An add operation of o=foo,c=uk on either Server A or Server B fails. Instead, Oracle Internet Directory returns a knowledge reference.

Configuring Default Knowledge References

Oracle Internet Directory uses the namingcontext attribute in the **directory-specific entry (DSE)** to determine all the **naming contexts** held locally by the server. Be sure that the namingContext attribute correctly reflects the naming context information.

You specify default knowledge references by entering a value for the ref attribute in the DSE entry. If the ref attribute is not in the DSE entry, then no default knowledge reference is returned.

When configuring a default knowledge reference, do not specify the DN in the LDAP URL.

For example, suppose that the DSE entry on Server A contains the following `namingContext` value:

```
namingcontext: c=us
```

Further, suppose that the default knowledge reference is:

```
Ref: ldap://host_PQR:389/
```

Now, suppose that a user enters an operation on Server A that has a base DN in the naming context `c=canada`, for example:

```
ou=marketing,o=foo,c=canada
```

This user would receive a knowledge reference to the host PQR. This is because Server A does not hold the `c=canada` base DN, and the `namingcontext` attribute in its DSE does not hold the value `c=canada`.

See Also: "About Knowledge References (Referrals)" on page 2-26 for a conceptual discussion of knowledge references

9

Managing Globalization Support in the Directory

Oracle Internet Directory uses Globalization Support to store, process and retrieve data in native languages. It ensures that Oracle Internet Directory utilities and error messages automatically adapt to the native language and locale.

This chapter discusses Globalization Support as used by Oracle Internet Directory and tells you the required NLS_LANG environment variables for the various components and tools in an Oracle Internet Directory environment.

> **See Also:** "Globalization Support" on page 2-14 prior to configuring Globalization Support

This chapter contains these topics:

- The NLS_LANG Environment Variable
- Using Globalization Support with LDIF Files
- Using Globalization Support with Command-Line Tools
- Setting NLS_LANG in the Client Environment
- Using Globalization Support with Bulk Tools

The NLS_LANG Environment Variable

The NLS_LANG parameter has three components—`language`, `territory`, and `charset`—in the form:

```
NLS_LANG = language_territory.charset
```

Each component controls the operation of a subset of Globalization Support features.

Component	Description
language	Specifies conventions such as the language used for Oracle messages, day names, and month names. Each supported language has a unique name—for example, American English, French, or German. The language argument specifies default values for the territory and character set arguments, so either (or both) `territory` or `charset` can be omitted. If language is not specified, the value defaults to American English. **See Also:** *Oracle9i Globalization and National Language Support Guide* for a complete list of languages
territory	Specifies conventions such as the default calendar, collation, date, monetary, and numeric formats. Each supported territory has a unique name; for example, America, France, or Canada. If territory is not specified, the value defaults to America. **See Also:** *Oracle9i Globalization and National Language Support Guide* for a complete list of territories
charset	Specifies the character set used by the client application (normally that of the user's terminal). Each supported character set has a unique acronym, for example, US7ASCII, WE8ISO8859P1, WE8DEC, WE8EBCDIC500, or JA16EUC. Each language has a default character set associated with it. Default values for the languages available on your system are listed in your operating system installation guide or administrator's guide. Oracle Internet Directory requires all data to be stored in UTF-8. **See Also:** *Oracle9i Globalization and National Language Support Guide* for a complete list of character sets

Note: All components of the NLS_LANG definition are optional, that is, any item left out will default.

Also, if you specify `territory` or `charset`, you *must* include the preceding delimiter [underscore (_) for `territory`, and period (.) for `charset`], otherwise the entire value will be parsed as a language name.

You can set NLS_LANG as an environment variable at the command line. The following are examples of legal values for NLS_LANG:

- `AMERICAN_AMERICA.UTF8`
- `JAPANESE_JAPAN.UTF8`

Using Globalization Support with LDIF Files

See Also: "LDAP Data Interchange Format (LDIF) Syntax" on page A-2

Attribute types are always ASCII strings that cannot contain multibyte characters. Oracle Internet Directory does not support multibyte characters in attribute type names. However, Oracle Internet Directory does support attribute *values* containing multibyte characters such as those in the simplified Chinese (.ZHS16GBK) character set.

Attribute values can be encoded in different ways to allow Oracle Internet Directory tools to interpret them properly. There are two scenarios:

- An LDIF file Containing Only ASCII Strings
- An LDIF file Containing UTF-8 Encoded Strings

An LDIF file Containing Only ASCII Strings

In this scenario, character strings for attribute values are also in ASCII.

Because all tools use the UTF-8 character set by default, and ASCII is a proper subset of UTF-8, all tools can interpret these files. The same is true of keyboard input of values that are simply ASCII strings.

An LDIF file Containing UTF-8 Encoded Strings

In this scenario, character strings for attribute values are also in UTF-8.

Because all tools use the UTF-8 character set by default, all tools can interpret these files. The same is true of keyboard input of values which are UTF-8 strings.

In such a file, some characters may be multibyte. Multibyte characters strings can be present in the LDIF files as attribute values or given as keyboard input. They can be encoded in their native character set or in UTF-8. They can also be BASE64 encoded representations of either the native or the UTF-8 string.

Consider the following cases:

- CASE 1: Native Strings (Non-UTF-8)
- CASE 2: UTF-8 Strings
- CASE 3: BASE64 Encoded UTF-8 Strings
- CASE 4: BASE64 Encoded Native Strings

Because the LDAP server understands and expects only UTF-8 encoded strings, cases 1, 3, and 4 need to undergo conversion to UTF-8 strings before they can be sent to the LDAP server.

CASE 1: Native Strings (Non-UTF-8)

Use the `-E` argument in the command-line tools, ldifwrite, and bulkmodify. Use the `-encode` argument in the bulkload and bulkdelete tools.

This example converts simplified Chinese native strings to UTF-8. The baseDN can be a simplified Chinese string:

```
ldapsearch -h my_host -p 389 -E ".ZHS16GBK" -b base_DN  -s base  "objectclass=*"
```

CASE 2: UTF-8 Strings

No conversion is required.

CASE 3: BASE64 Encoded UTF-8 Strings

You need to use neither the `-E` argument in the command-line tools, ldifwrite, and bulkmodify, nor the `-encode` argument in bulkload and bulkdelete. Oracle Internet Directory tools automatically decode BASE64 encoded UTF-8 strings to UTF-8 strings.

CASE 4: BASE64 Encoded Native Strings

Use the `-E` argument in the command-line tools, ldifwrite, and bulkmodify. Use the `-encode` argument in the bulkload and bulkdelete tools.

Oracle Internet Directory tools automatically decode BASE64 encoded native strings to simple native strings. The native strings are then converted to the equivalent UTF-8 strings.

> **Note:** In any given input file, only one language set may be used.

Using Globalization Support with Command-Line Tools

The Oracle Internet Directory command-line tools read keyboard input or LDIF file input in the following ways:

- ASCII characters only
- Non-ASCII input (native language character set)
- BASE64 encoded values of UTF-8 or native strings (from LDIF file only)

If the character set being given as input from an LDIF file or keyboard is not UTF-8, the command-line tools need to convert the input into UTF-8 format before sending it to the LDAP server.

You enable the command-line tools to convert the input into UTF-8 by specifying the `-E` argument when using each tool.

This section contains these topics:

- Specifying the -E Argument When Using Each Tool
- Examples: Using the -E Argument with Command-Line Tools

Specifying the -E Argument When Using Each Tool

The client tools always assume UTF-8 to be the character set unless otherwise specified by the `-E` argument. The BASE64-encoded values are decoded, and then the decoded buffer is converted to UTF-8 if the `-E` argument is specified. For example, if you specify `-E ".ZHS16GBK"`, then the decoded buffer is converted from simplified Chinese to UTF-8 before being sent to the LDAP server.

Specifying the `-E` argument ensures that proper character set conversion can occur from the character set you specify for the `-E` argument (`-E ".`*`character_set`*`"`) to the.UTF-8 character set.

The command-line tools use the `-E` argument to process the input in the character set specified for the `-E` argument. They display their output in the character set specified in the NLS_LANG environment variable.

For example, to add entries from an LDIF file encoded in the simplified Chinese character set (.ZHS16GBK) by using ldapadd, type:

```
ldapadd -h myhost -p 389 -E ".ZHS16GBK" -f my_ldif_file
```

In this example, the ldapadd tool converts the characters from `".ZHS16GBK"` (simplified Chinese character set) to "`.UTF8`" (UTF-8 character set) before they are sent across the wire to the LDAP server.

Examples: Using the -E Argument with Command-Line Tools

The following table provides additional examples of how to use the `-E` argument correctly for each command-line tool. In each example, the command converts data from simplified Chinese, as specified by the value `".ZHS16GBK"`, to UTF-8. For example, in each command, the values for the `-D` and `-w` options are in simplified Chinese. Specifying the `-E` argument converts them to UTF-8.

Note that, in the examples in the following table, we do not show any actual characters belonging to `.ZHS16GBK` character set. These examples would, therefore, work without the `-E` argument. However, if the argument values contained actual characters in the `.ZHS16GBK` character set, then we would need to use the `-E` argument.

> **See Also:** Appendix A, "Syntax for LDIF and Command-Line Tools" for syntax and usage notes for each of the command-line tools

Tool	Example

ldapbind	`ldapbind -h my_host -p 389 -E ".ZHS16GBK"`
	`-D "o=acme,c=us" -w my_password`

ldapsearch	`ldapsearch -h my_host -p 389 -E ".ZHS16GBK"`
	`-D "o=acme,c=us" -w my_password`

ldapadd	`ldapadd -h my_host -p 389 -E ".ZHS16GBK"`
	`-D "o=acme,c=us" -w my_password`

ldapaddmt	`ldapaddmt -h my_host -p 389 -E ".ZHS16GBK"`
	`-D "o=acme,c=us" -w my_password`

Tool	Example
ldapmodify	`ldapmodify -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password`
ldapmodifymt	`ldapmodifymt -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password`
ldapdelete	`ldapdelete -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password`
ldapcompare	`ldapcompare -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password -b "ou=Construction,ou=Manufacturing,o=acme,c=us" -a title -v manager`
ldapmoddn	`ldapmoddn -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password -b "cn=Franklin Badlwins,ou=Construction,ou=Manufacturing,c=us,o=acme" -N "ou=Contracting,ou=Manufacturing,o=acme,c=us" -r`

Setting NLS_LANG in the Client Environment

If the output required by the client is UTF-8, then you do not need to set the NLS_LANG environment variable. In this case, the NLS_LANG environment variable defaults to `.UTF8`, and both the input path from client to server, and the output path from server to client, do not require any character set conversion.

If the output required by the client is *not* UTF-8, then you must set the NLS_LANG environment variable. This ensures that proper character set conversion can occur from the UTF-8 character set to the character set required by the client.

For example, if the NLS_LANG environment variable is set to the simplified Chinese character set, then the command-line tool displays output in that character set. Otherwise the output defaults to the UTF-8 character set.

Note: If you are using Windows NT, then, to use the command-line tools after server startup, you must reset NLS_LANG in an MS-DOS window. Set it to the character set that matches the code page of your MS-DOS session. (UTF-8 cannot be used.) See the *Oracle9i Database Installation Guide for Windows* for more information on which character set to use for command-line tools in an MS-DOS session.

If you are using a pre-installed Oracle9*i* release 9.0.1 database with Oracle Internet Directory, then you must also set the database character set to UTF-8. See the *Oracle9i Globalization and National Language Support Guide* and *Oracle9i Database Installation Guide for Windows* for more information.

Be careful not to change the NLS_LANG parameter value in the registry.

Using Globalization Support with Bulk Tools

Oracle Internet Directory ensures that the reading and writing of text data from and to LDIF files are done in UTF-8 encoding as specified by the LDAP standard.

This section provides an example of the argument you use for each of the following bulk tools:

- Using Globalization Support with bulkload
- Using Globalization Support with ldifwrite
- Using Globalization Support with bulkdelete
- Using Globalization Support with bulkmodify

See Also: "Bulk Tools Syntax" for a list of arguments for each bulk tool

Using Globalization Support with bulkload

Add to the command the argument `-encode "`*`character_set`*`"` where the input LDIF file is encoded in *"character_set"*.

For example:

```
bulkload.sh -connect net_service_name -encode ".ZHS16GBK" my_ldif_file
```

Using Globalization Support with ldifwrite

The ldifwrite utility always writes BASE64 encoded values for multibyte strings.

The BASE64 encoding could be of the UTF-8 strings as they are stored in the directory server, or of native strings as specified by the NLS_LANG environment variable setting when running ldifwrite.

For example:

```
ldifwrite -c net_service_name -b baseDN -f output_file
```

In this example, if the NLS_LANG environment variable is not set, or is set to *language_territory*.UTF8, then the output LDIF file will contain BASE64-encoded UTF-8 strings for any multibyte characters.

To reload this LDIF file into the directory by using ldapaddmt, use the following syntax:

```
ldapaddmt -h my_host -p port_number -f output_file
```

In the above case, the -E argument is not required because the decoded BASE64 strings are already UTF-8-encoded and can be readily sent to the server.

If the NLS_LANG environment variable is set to a character set other than UTF-8—for example, ".ZHS16GBK"—then the output LDIF file will contain a BASE64 encoded value of simplified Chinese (.ZHS16GBK) strings.

To reload this LDIF file into the directory using ldapaddmt, use the following syntax:

```
ldapaddmt -h host -p port -E ".ZHS16GBK" -f my_input_file.LDIF
```

In the above case the -E argument is required because the decoded BASE64 strings are simplified Chinese, which need to be converted to UTF-8 strings before being sent to the server.

Using Globalization Support with bulkdelete

Add -encode ".*character_set*" to the command.

For example:

```
bulkdelete.sh -connect net_service_name -encode ".ZHS16GBK" -base
"ou=manufacturing,o=acme,c=us"
```

In this case the value for the -base option could be in the ZHS16GBK native character set, that is, simplified Chinese.

Using Globalization Support with bulkmodify

Add -E ".*character_set*" to the command the argument.

For example:

```
bulkmodify.sh -c my_service_name -E ".ZHS16GBK" -b
"ou=manufacturing,o=acme,c=us" -r title -v Foreman -f "objectclass=*"
```

In this example, values for the -b, -v, and -f arguments can be specified using the simplified Chinese character set.

10

Managing the Delegated Administration Service

The Delegated Administration Service enables directory users to modify their own personal data—such as addresses, phone numbers, and photos—without the intervention of an administrator. It also enables users to search other parts of the directory to which they have access. This frees directory administrators for other tasks in the enterprise.

This chapter contains these topics:

- Concepts and Architecture
- Starting and Stopping the Apache Server
- Installing and Configuring the Delegated Administration Service

> **See Also:** The online help for the Delegated Administration Service for instructions on how to modify directory data by using this service

Concepts and Architecture

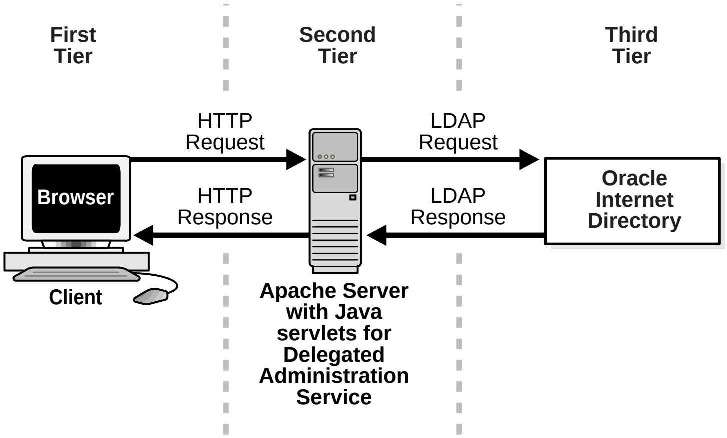
The Delegated Administration Service relies on a Web server, that is, a program that delivers Web pages. More specifically, it uses an Apache Web server, one of the most widely used Web servers.

The Apache Web server is enabled for small Java programs, called servlets. Together, the Apache Web server and the servlets do the following:

1. Receive requests from clients
2. Process those requests—by either retrieving or updating data in Oracle Internet Directory—then generate results
3. Send responses back to clients

Figure 10–1 shows the relationship between components of the Delegated Administration Service.

Figure 10–1 Components of the Delegated Administration Service


First Tier
Second Tier
Third Tier
HTTP Request
HTTP Response
LDAP Request
LDAP Response
Browser
Client
Oracle Internet Directory
Apache Server with Java servlets for Delegated Administration Service

In the first tier, the user sends to the Apache server an HTTP request containing a query to Oracle Internet Directory.

In the second tier, the Apache server receives the request and launches the appropriate Delegated Administration Service servlet. The Delegated Administration Service servlet interprets the request, and sends it Oracle Internet Directory on the third tier.

After the Delegated Administration Service servlet receives the LDAP result from Oracle Internet Directory, it compiles that result into an HTML page, and sends it back to the client Web browser.

Starting and Stopping the Apache Server

Start the Apache server by entering:

```
$ORACLE_HOME/Apache/Apache/bin/apachectl start
```

Stop the Apache server by entering:

```
$ORACLE_HOME/Apache/Apache/bin/apachectl stop
```

Installing and Configuring the Delegated Administration Service

To install and configure the Delegated Administration Service, perform these tasks:

- Task 1: Install the Delegated Administration Service
- Task 2: Configure the Delegated Administration Service
- Task 3: Verify that the Delegated Administration Service Is Running

Task 1: Install the Delegated Administration Service

The Delegated Administration Service is installed along with Oracle Internet Directory release 3.0.1. If you want to enable Single Sign-On, then you must install and configure the login server.

See Also:

- Installation documentation for Oracle Internet Directory release 3.0.1 for your operating system
- *Single Sign-On Administrator's Guide*

Task 2: Configure the Delegated Administration Service

To configure the Delegated Administration Service, use a text editor to modify parameters in the `oidprefs.properties` file located in the *`ORACLE_HOME`*`/ldap/ssa` directory. The following sections discuss the parameters in that file.

The log file location for the Delegated Administration Service is located at `$ORACLE_HOME/ldap/ssa/logs/ssa.log`.

General Parameters

The Delegated Administration Service uses a special account to initialize and reset user passwords. If you are using IMAP authentication described in "Parameters for Registering and Resetting Passwords" on page 10-6, then you need to configure this special account to initialize and reset user passwords. To do this, run the script setup_admin.sh in the directory `$ORACLE_HOME/ldap/ssa`. This script creates the default special administrator account and sets the privileges for it.

If Single Sign-On is enabled, then the Delegated Administration Service uses the Oracle Internet Directory proxy user feature. To use Single Sign-On, configure the parameters for the proxy user in Table 10–1.

Table 10–1 explains the fields for setting general parameters in the `oidprefs.properties` file:

Table 10–1 General Parameters in the oidprefs.properties File

Entry	Description
`oidhost`	Enter the fully qualified host name where the directory server is running and which you are using with the Delegated Administration Service. There is no default.
`corproot`	Enter the corporation root entry. Modify this field to comply your deployment environment. All user entries must exist below this container. The default is `dc=oracle, dc=com`.
`loginnameattr`	Enter the attribute that stores the user login identifier. This attribute needs to be indexed. It should uniquely identify the user in the organization under the specified corporation root. The default is `uid`.
`Mailinglistobjectclass`	Enter the object class that contains the mailing list-specific attributes. The default is `mailgroup`.
`employeeobjectclass`	Enter the object class that contains the user specific attributes. The default is `orclmailuser`.
`ssadebug`	Enable or disable debug logging for the Delegated Administration Service. The default is `True`. To disable debugging, set this value to `False`.
`ssahostport`	Point this entry to the following URL: `http://`*`your_host`*`:`*`http_port`*. The default is `http://`*`local_host_name`*`:7777`

Table 10–1 General Parameters in the oidprefs.properties File

Entry	Description
`oidacct`	Enter the DN of the administration account for the user password population. This is used to populate user password for Oracle Internet Directory registration. You configure this account by running the script `setup_admin.sh` in the directory `$`*`ORACLE_HOME`*`/ldap/ssa`. The default value is `cn=oidpasswordadmin,dc=oracle,dc=com`.
`oidpwd`	Enter the password of the administration account specified in the `oidacct` configuration field. The default value is `welcome`.
`proxydn`	Within a Single Sign-On environment enabling the Delegated Administration Service, enter the DN for the proxy account used to switch the initial LDAP proxy connection to the login user connection. The default value is `cn=proxy`.
`proxypwd`	Enter the password of the proxy account. The default value is `proxy`.
`serverloc`	Enter the Apache image directory, that is, the local file system directory where the Apache server stores the images retrieved from the directory server to make them accessible to all HTTP connections to the Delegated Administration Service. For the Oracle Portal platform, it is located at `$`*`ORACLE_HOME`*`/webdb30/images`.
`passwordpolicyrule`	Customize the password policy. You can enforce the minimum password length and the number of letters and numerals. The default is `len:5:letter:1:numeric:1.` **See Also:** "Password Policies" on page 11-7 for a conceptual discussion of password policies

Note: Once you have modified the `oidprefs.properties` file, you must stop, then restart, the Apache server for your changes to take effect.

Parameters for Registering and Resetting Passwords

To enable users to self-register and reset their passwords, you configure these properties. In release 3.0.1, the Delegated Administration Service verifies user credentials by using IMAP authentication only. You may use this if you have an IMAP server and want to use it to authenticate users.

The link `(initial registration/forgot password)` on the oidprefs login page:

- Takes you to the page that checks your IMAP credentials
- Sets the Oracle Internet Directory user password to the IMAP password.

If you do not want to use this feature, point the `resetpasswordurl` parameter to an HTML page with instructions for users to register or reset their passwords.

Table 10–2 Parameters for Registration and Resetting Passwords in the oidprefs.properties File

Entry	Description
`emailserver`	To enable self-registration of users, enter the fully qualified host name of your organization IMAP server. There is no default.
`emailport`	Enter the IMAP server port. The default is 143.
`resetpasswordurl`	If you have an IMAP server, and you want to enable self-registration, then use the default value, namely, `/servlet/imAuth`. Otherwise, customize the default value to point to another URL that provides this ability to users.

Parameters for Integrating with Single Sign-On

Table 10–3 explains the parameters you set in order to integrate the Delegated Administration Service with Single Sign-On.

Table 10–3 Parameters for Integrating with Single Sign-On in the oidprefs.properties File

Entry	Description
`ssoenabled`	Enable or disable Single Sign-On. The default is `False`.
`ssopwdchange`	Enable or disable usage of the Single Sign-On password change page. If it is disabled, then the Delegated Administration Service uses its own password change page.

Table 10–3 Parameters for Integrating with Single Sign-On in the oidprefs.properties File

Entry	Description
`tokenurl`	Enter the token used to register the Delegated Administration Service as a partner application within a Single Sign-On environment. If `ssoenabled` is set to False, then this field is displayed as empty.
`ssookurl`	Enter the page that appears after the user clicks OK on the password change page in Single Sign-On. If `ssoenabled` is set to false, then this field is displayed as empty.
`ssocancelurl`	Specify the URL for the HTML page that appears after the user clicks Cancel button on the Single Sign-On password change page. If `ssoenabled` is set to False, then this field is displayed as empty.
`oidpartnerid`	Enter the identifier stored in the cookie for login user. The default is `OID_PARTNER_ID`. Do not modify this parameter.
`ssodbuser`	Enter the user identifier for JDBC connection to the Single Sign-On database. If `ssoenabled` is set to false, then this field is displayed as empty.
`ssodbpwd`	Enter the password for JDBC connection to the Single Sign-On database. If `ssoenabled` is set to false, then this field is displayed as empty.
`ssodbhost`	Enter the name of the host for the Single Sign-On database. You need to modify this field. If `ssoenabled` is set to false, then this field is displayed as empty.
`ssodbport`	Enter the number of the Single Sign-On database port. If `ssoenabled` is set to false, then this field is displayed as empty.
`ssodbsid`	Enter the Single Sign-On database SID. If `ssoenabled` is set to false, then this field is displayed as empty.
`ssourl`	Enter the URL for the HTML page that appears after the user clicks the OK button on the Single Sign-On password change page. The value must be in this format: `http://Apache_server_host>:`*`port_number/`*`servlet/root`. For example, if the Apache server is running on `My_computer`, and the port number is 7777, then the value you enter is `http://My_computer:7777/servlet/root`

Task 3: Verify that the Delegated Administration Service Is Running

To do this, follow these steps:

Step 1: Verify that the Apache Server Is Running

To do this, check the log files for the Apache server. Enter:

```
ps -ef | grep http
```

This command generates the related log files under the following directories:

Application	Log File Location
Apache Server	`$ORACLE_HOME/Apache/Apache/logs`
Java Servlets for the Delegated Administration Service	`$ORACLE_HOME/Apache/Jserv/logs`
Delegated Administration Service	`$ORACLE_HOME/ldap/ssa/logs`

See Also: "Starting and Stopping the Apache Server" on page 10-3

Step 2: Verify that the Delegated Administration Service Is Running

Using any browser, enter:

```
http://host_name:7777/servlets/oidprefs
```

where *host_name* is the name of the computer on which the Apache server is running. This displays the Delegated Administration Service logon screen.

Part III

Directory Security

This part contains discusses the features that enable you to secure data within the directory, as well as how to establish access controls for administering applications in enterprises and hosted environments. It contains these chapters:

11

About Security in Oracle Internet Directory

This chapter describes the security features available with Oracle Internet Directory, and explains how to deploy the directory for administrative delegation. It contains these topics:

- Security Features of Oracle Internet Directory
- Directory-Based Application Security

Security Features of Oracle Internet Directory

This section describes each Oracle Internet Directory security feature. It contains these topics:

- Data Integrity
- Data Privacy
- Authentication
- Authorization
- Password Protection
- Password Policies

Data Integrity

Oracle Internet Directory ensures that data has not been modified, deleted, or replayed during transmission by using Secure Sockets Layer (SSL). This SSL feature generates a cryptographically secure message digest—through cryptographic checksums using either the **MD5** algorithm or the **Secure Hash Algorithm (SHA)**—and includes it with each packet sent across the network.

> **See Also:** Chapter 12, "Managing Secure Sockets Layer (SSL)" for more information about SSL

Data Privacy

Oracle Internet Directory ensures that data is not disclosed during transmission by using **public-key encryption** available with SSL. In public-key encryption, the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the recipient decrypts the message using the recipient's private key. Specifically, Oracle Internet Directory supports two levels of encryption available through SSL:

- DES40

  The DES40 algorithm, available internationally, is a variant of **DES** in which the secret key is preprocessed to provide forty effective **key** bits. It is designed for use by customers outside the USA and Canada who want to use a DES-based encryption algorithm. This feature gives commercial customers a choice in the algorithm they use, regardless of their geographic location.

- RC4_40

  Oracle has obtained license to export the RC4 data encryption algorithm with a 40-bit key size to virtually all destinations where other Oracle products are available. This makes it possible for international corporations to safeguard their entire operations with fast cryptography.

  > **See Also:** Chapter 12, "Managing Secure Sockets Layer (SSL)" for more information about SSL

Authentication

Authentication is the process by which the directory server establishes the true identity of the user connecting to the directory. It occurs when an LDAP session is established by means of the ldapbind operation. Thus every session has an associated user identity.

To verify the identities of users, hosts, and clients, Oracle Internet Directory provides four authentication options:

Anonymous Authentication

When users authenticate anonymously, they simply leave the user name and password fields blank when they log in. Each anonymous user then exercises whatever privileges are specified for anonymous users.

Simple Authentication

When using simple authentication, the client identifies itself to the server by means of a DN and a password that are not encrypted when sent over the network.

Secure Sockets Layer (SSL) Authentication

This involves the exchange of certificate**s** issued by trusted certificate authorities.

Authentication Through a Middle Tier

Authentication through a middle tier, such as a RADIUS server or an LDAP self-service servlet, involves a proxy user that performs directory operations on the end user’s behalf. Authentication through a middle tier takes place as follows:

1. The end user authenticates to the middle tier.
2. The middle tier binds to the directory as a proxy user.

3. The proxy user performs a second bind, this time using the DN of the end user. It does not need to enter the end user's password.
4. The directory server recognizes this second bind as an attempt by the proxy user to switch to the end user's identity. The directory server trusts the authentication granted to the end user by the middle tier and allows this second bind to succeed.

The access controls that the Oracle directory server uses throughout the rest of the session are those it would use if the end user were performing those operations.

For example, suppose you have a proxy user whose DN is `cn=ProxyUser`. The middle tier service authenticates the end user. The middle tier service then binds to the directory as `cn=ProxyUser`. The proxy user then performs another bind, this time using the DN of the end user—let's call it `cn=EndUser`. The Oracle directory server recognizes this second bind as an attempt by the proxy user to switch its identity to `cn=EndUser` for the duration of the session. Because Oracle directory server trusts the proxy user, it allows this second bind to succeed; it does not require any further validation of the end-user DN, such as a password. For the rest of the session, all LDAP operations are access controlled as if `cn=EndUser` were performing them.

Authorization

Authorization is the process of ensuring that a user reads or updates only the information for which that user has privileges. When directory operations are attempted within a directory session, the directory server ensures that the user has the requisite permissions to perform those operations. If the user does not have the requisite permissions, then the directory server disallows the operation. Through this mechanism, the directory server protects directory data from unauthorized operations by directory users. This mechanism is called access control.

Access control information is the directory metadata that captures the administrative policies relating to access control. This information is stored in Oracle Internet Directory as user-modifiable operational attributes, each of which is called an **access control item (ACI)**.

Typically, a list of these ACI attribute values, called an **access control list (ACL)**, is associated with directory objects. The attribute values on that list govern the access policies for those directory objects.

Access control information associated with a directory object represents the permissions on the given object that various directory user entities (or subjects) have. Thus, an ACI consists of:

- The object to which you are granting access
- The entities or subjects to whom you are granting access
- The kind of access you are granting

Access control policies can be prescriptive, that is, their security directives can be set to apply downward to all entries at lower positions in the **directory information tree (DIT)**. The points from which such access control policies apply are called **access control policy points** (**ACPs**).

ACIs are represented and stored as text strings in the directory. These strings must conform to a well defined format, called the ACI directive format. Each valid value of an ACI attribute represents a distinct access control policy.

The following features of directory access control can be used by applications running in a hosted environment.

Prescriptive access control	Enables the service provider to specify access control lists (ACLs) for a collection of directory objects, instead of having to state the policies for each individual object. This feature simplifies the administration of access control, especially in large directories where many objects are governed by identical or similar policies.
Hierarchical access control administration model	Enables the service provider to delegate directory administration to subscribers. The subscriber could in turn delegate further if necessary.
Administrative override control for delegated domains	Enables the service provider to perform diagnosis and recovery from unintentional account lockout or accidental security exposure.

Dynamic evaluation of access control entities	Enables subtree administrators to identify both subjects and objects in terms of their namespace and their association with other objects in the directory. For example, the administrator of one subscriber subtree can allow only a user's manager to update that user's salary attribute. The administrator of another subscriber subtree can establish and enforce a different policy regarding salary attributes.

Password Protection

Oracle Internet Directory can protect passwords by storing them in the `userPassword` attribute as one-way hashed values. You select the hashing algorithm you want to use. Storing passwords as one-way hashed values—rather than as encrypted values—more fully secures them because a malicious user can neither read nor decrypt them.

During authentication to a directory server, a user enters a password in clear text. The directory server hashes this user password by using the specified hashing algorithm, then verifies it against the hashed password stored in the `userPassword` attribute. If the hashed password values match, then the server authenticates the user. If they do not match, then the server sends the user an Invalid Credentials error message.

You can specify one of the following hashing schemes:

- **MD4**—The default hashing scheme. It is a one-way hash function that produces a 128-bit hash, or message digest
- **MD5**—An improved, and more complex, version of MD4
- **SHA**—Secure Hash Algorithm, which produces a 160-bit hash, longer than MD5. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.
- **UNIX Crypt**—The UNIX hashing algorithm
- No Hashing

The hashing algorithm value you specify is stored in the `orclCryptoScheme` attribute in the **root DSE**. This attribute is single-valued.

See Also: "Managing Password Protection" on page 6-21

Password Policies

A password policy is a set of rules governing how passwords are used. When a user attempts to bind to the directory, the directory server ensures that the password meets the various requirements set in the password policy.

When you establish a password policy, you set the following types of rules, to mention just a few:

- The maximum length of time a given password is valid
- The minimum number of characters a password must contain
- The ability of users to change their own passwords

> **See Also:** "Managing Password Policies" on page 6-17 for a fuller description of the rules you set when establishing password policies

Directory-Based Application Security

Because directory access control policies are stored as LDAP attributes, you can set metapolicies controlling who can modify them. This enables a global administrator to assign privileges to administrators of specific subtrees—for example, to administrators of applications in a hosted environment. Similarly, a global administrator can delegate to departmental administrators access to the metadata of applications in their departments. Department administrators can then control access to their department applications.

Thus, you can implement access control on two levels:

Authorization of users

In this case, the directory stores access control policies that external applications then read and enforce. When a user tries to perform an operation by using an application, the application verifies that the user has the correct authorization to perform the operation.

Authorization of administrators

In this case, the directory serves as the trusted point of administration for all application-specific access control polices. To govern who can administer the access control policies of specific applications, you set access control policies at the directory level for these applications. Then, when a user attempts to change an application-specific access control policy, the directory verifies that the user has the correct authorization to make that change.

Figure 11–1 shows the relationship between directory access control and the application-specific access control mechanisms in a hosted environment.

Figure 11–1 Directory Access Control and Application-Specific Access Control

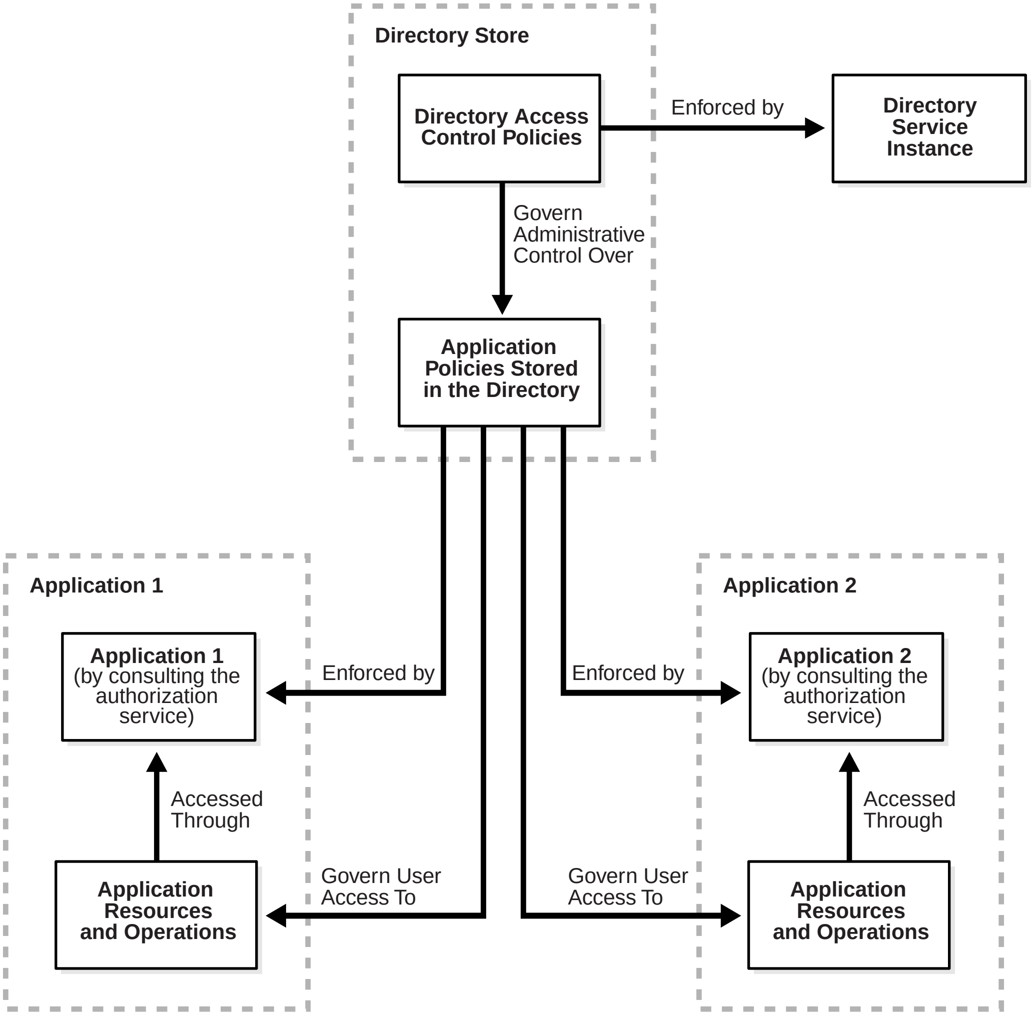
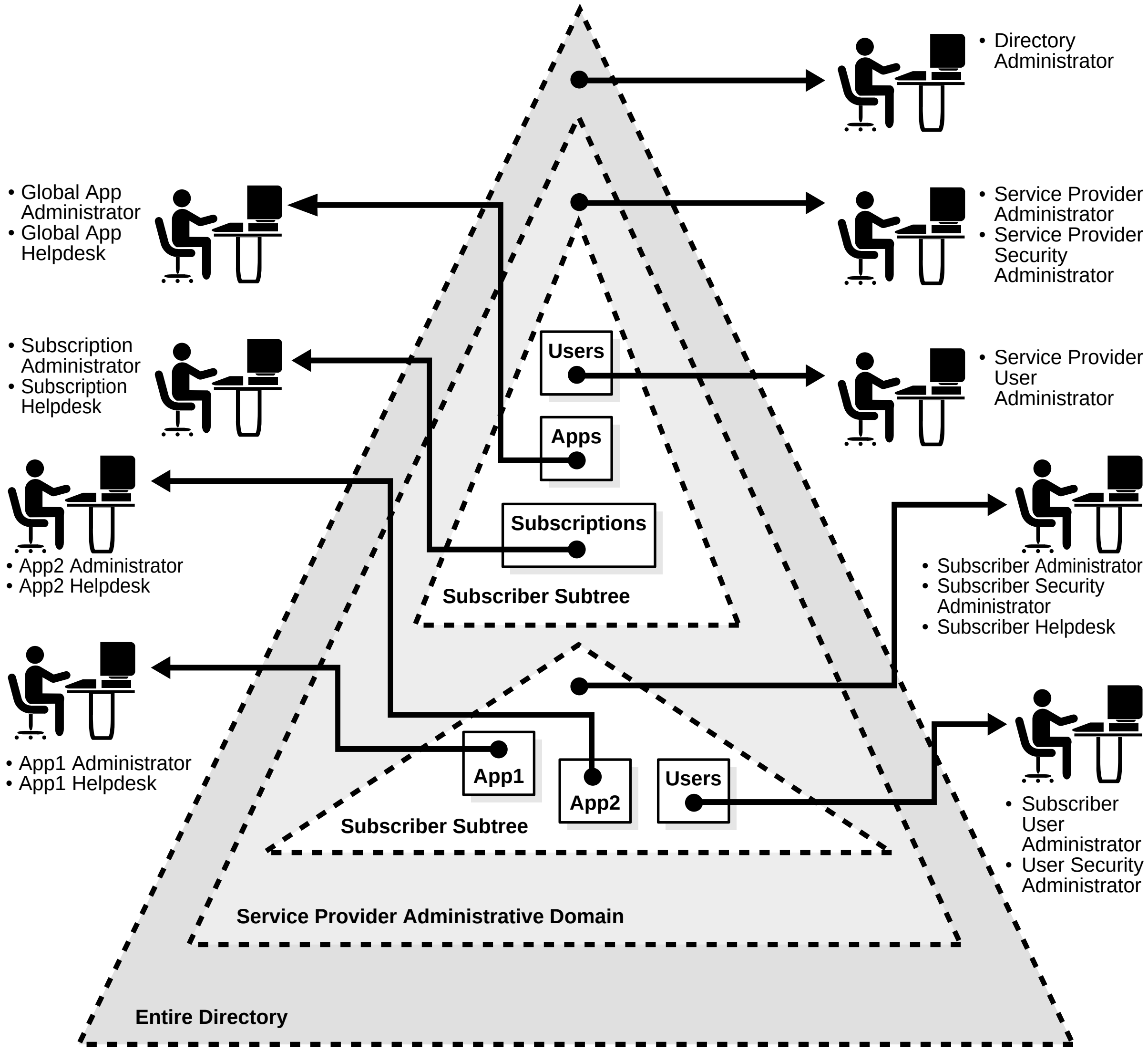

Directory Store
Directory Access Control Policies
Enforced by
Directory Service Instance
Govern Administrative Control Over
Application Policies Stored in the Directory
Application 1
Application 1 (by consulting the authorization service)
Enforced by
Accessed Through
Application Resources and Operations
Govern User Access To
Application 2
Application 2 (by consulting the authorization service)
Enforced by
Accessed Through
Application Resources and Operations
Govern User Access To

Figure 11–2 illustrates the various domains and the roles associated with them in the directory.

Figure 11–2 Directory Domains and Roles in a Hosted Environment


• Directory Administrator
• Global App Administrator
• Global App Helpdesk
• Service Provider Administrator
• Service Provider Security Administrator
Users
• Subscription Administrator
• Subscription Helpdesk
• Service Provider User Administrator
Apps
Subscriptions
• App2 Administrator
• App2 Helpdesk
Subscriber Subtree
• Subscriber Administrator
• Subscriber Security Administrator
• Subscriber Helpdesk
• App1 Administrator
• App1 Helpdesk
App1
App2
Users
Subscriber Subtree
• Subscriber User Administrator
• User Security Administrator
Service Provider Administrative Domain
Entire Directory

In Figure 11–2, each triangle represents a portion of a DIT.

- The outermost triangle represents the entire directory. The directory administrator has privileges extending across the entire directory.
- Immediately inside the outermost triangle, another triangle represents the service provider administrative domain. In this domain, privileges to add new entries are delegated to the service provider's administrators.
- Inside the service provider administrative domain, privileges can be further delegated based on the ownership of directory information. For example, the delegation can depend on whether the information is private to a specific subscriber or global to the service provider.

Figure 11–2 shows only a single subscriber represented in the directory. In reality there are multiple subscribers, each with its own domain requiring protection from the others.

Some of the protection domains in this model are:

- Entire directory
- Service provider administrative domain
- Service provider-specific directory information tree
- Subscriber-specific subtree
- Application-specific footprint in the directory
- User-specific information

These protection domains are supported by the following roles, which enable the service provider or subscriber to customize access control.

Global Administrative Roles
These roles have rights to perform activities that span the entire directory.

Subscriber-Specific Roles
These roles are limited to the directory trees specific to the subscribers.

Application-Specific Roles

When hosting directory-enabled applications, it is not necessary to represent all application-specific roles in the directory. However, it is better that applications, when representing roles that directly affect their directory footprint, follow the delegation model recommendations described earlier. This enables applications to leverage the directory-based delegation model when granting directory-specific privileges to users.

12

Managing Secure Sockets Layer (SSL)

This chapter explains how to configure Secure Sockets Layer (SSL) for use with Oracle Internet Directory. If you use Secure Sockets Layer (SSL), you may also configure strong authentication, data integrity, and data privacy.

This chapter contains these topics:

- Supported Cipher Suites
- SSL Client Scenarios
- Configuring SSL Parameters
- Issues Specific to This Release of Oracle Internet Directory

> **See Also:** "Security" on page 2-13 for a conceptual overview of SSL in relation to Oracle Internet Directory

Supported Cipher Suites

A cipher suite is a set of authentication, encryption, and data integrity algorithms used for exchanging messages between network nodes. During an SSL handshake, the two nodes negotiate to see which cipher suite they will use when transmitting messages back and forth.

The Oracle Internet Directory supports the following SSL cipher suites:

Table 12–1 SSL Cipher Suites Supported in Oracle Internet Directory

Cipher Suite	Authentication	Encryption	Data Integrity
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	RSA	DES40	SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5	RSA	RC4_40	MD5
SSL_RSA_WITH_NULL_SHA	RSA	None	SHA
SSL_RSA_WITH_NULL_MD5	RSA	None	MD5

SSL Client Scenarios

Oracle Internet Directory clients can use SSL 2.0 or SSL 3.0. A client over SSL can connect to a server anonymously or by using either simple or strong authentication.

When both a client and server authenticate themselves to each other, SSL derives the identity information it requires from the X509v3 digital certificates.

Configuring SSL Parameters

During start-up of a directory **directory server instance**, the directory reads a set of configuration parameters, including the parameters for the SSL profile. If you are going to run the directory with SSL enabled, you need to examine—and possibly reconfigure—the SSL parameters in the **configuration set entry**.

To run a server instance in secure mode, modify the configuration settings to run with the secure port 636 as the default port.

You can create and modify multiple sets of configuration parameters with differing values, using a different configuration set entry for each instance of Oracle Internet Directory. This is a useful way to accommodate clients with different security needs.

Oracle Corporation recommends that you create separate configuration sets and modify their SSL values, rather than modify SSL values in the default configuration set. This is because the default configuration set may be required by Oracle Support Services in the diagnosis of certain technical issues.

See Also:

- "Managing Server Configuration Set Entries" on page 6-2 for instructions on how to set these parameters
- "Configuration Set Entry Attributes" on page C-5 for a description of these parameters

Configuring SSL Parameters by Using Oracle Directory Manager

You can examine and modify the values for the SSL configuration parameters in each configuration set entry that you have created and in each server instance that is currently running.

Note: You cannot directly change the parameters for an active instance. If you want to change the parameters for an active instance, change the parameters in a configuration set entry and save it. After it is saved, you can stop current instances and refer to the newly modified configuration set in the start server message.

To view and modify SSL configuration parameters:

1. In Oracle Directory Manager's navigator pane, expand Oracle Internet Directory Servers > *directory server* > Server Management.
2. Expand either Directory Server or Replication Server, as appropriate. The numbered configuration sets are listed beneath your selection.
3. Select the configuration set that you want to examine. The group of tab pages for that configuration set entry appear in the right pane.
4. Select the SSL Settings tab page.

You can change the parameters in this tab page and save them. The fields in this tab page are described in the following table:

Field	Description
SSL Enable	Select to enable SSL authentication. If you do not select this check box, SSL is not enabled, and you do not need to set any other parameters on this page.
SSL Authentication	Choose one of the following: ■ No SSL Authentication—Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, SSL encryption / decryption only is used. ■ SSL Client and Server Authentication—Both client and server authenticate themselves to each other and send certificates to each other. ■ SSL Server Authentication—Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.

SSL Wallet URL	Type the location of the server-side SSL wallet. If you elect to change the location of the wallet, you must change this parameter. You must set the wallet location on both the client and the server. For example, on Solaris, you could set this parameter as follows:
	`file:/home/my_dir/my_wallet`
	On Windows NT, you could set this parameter as follows:
	`file:C:\my_dir\my_wallet`

SSL Wallet Password	Type the password for the server-side wallet. This password was set during creation of the wallet. If you change the password, you must change this parameter.
SSL Wallet Confirm Password	Retype the new password in this field when you change the password.
SSL Port	The default SSL port is 636. You can change the SSL port.

See Also: "Managing Server Configuration Set Entries by Using Oracle Directory Manager" on page 6-4 for information about changing parameters in a configuration set entry

Configuring SSL Parameters by Using Command-Line Tools

See Also: "Managing Server Configuration Set Entries by Using Command-Line Tools" on page 6-10

Issues Specific to This Release of Oracle Internet Directory

If you intend to support both SSL and non-SSL clients on the same host, you need to configure two distinct server instances.

In Oracle Internet Directory release 3.0.1, the Oracle directory replication server cannot communicate directly with SSL-enabled Oracle directory server instances.

See Also: Chapter 6, "Managing the Oracle Directory Server" for instructions on how to configure server instances

13

Managing Directory Access Control

This chapter provides an overview of access control policies and describes how to administer directory access control by using either Oracle Directory Manager or the command-line tool, ldapmodify.

This chapter contains these topics:

- Overview of Access Control Policy Administration
- Managing Access Control by Using Oracle Directory Manager
- Managing Access Control by Using Command-Line Tools

See Also:

- "Globalization Support" on page 2-14 for a conceptual explanation before you begin implementing and administering access control policies
- Appendix B, "Using Access Control Directive Format" for information about the format or syntax of Access Control Items (ACIs)

Overview of Access Control Policy Administration

You manage access control policies by configuring the values of the **ACI** attributes within appropriate entries. You can do this by using either Oracle Directory Manager or ldapmodify.

This section contains these topics:

- Access Control Management Constructs
- Access Control Information Components
- How ACL Evaluation Works

Access Control Management Constructs

This section discusses the structures used for access control in Oracle Internet Directory. These include:

- Access Control Policy Points (ACPs)
- orclACI
- orclEntryLevelACI
- Privilege Groups

Access Control Policy Points (ACPs)

ACPs are entries in which the `orclACI` attribute has been given a value. The `orclACI` attribute value represents the access policies that are inherited by the subtree of entries starting with the ACP as the root of the subtree.

When a hierarchy of multiple ACPs exists in a directory subtree, a subordinate entry in that subtree inherits the access policies from all of the superior ACPs. The resulting policy is an aggregation of the policies within the ACP hierarchy above the entry.

For example, if an ACP is established in the HR department entry, and the Benefits, Payroll, and Insurance groups are entries within the HR department, then any entry within those groups inherits the access rights specified in the HR department entry.

When there are conflicting policies within a hierarchy of ACPs, the directory applies well-defined precedence rules in evaluating the aggregate policy.

See Also: "How ACL Evaluation Works" on page 13-10

orclACI

The orclACI attribute contains **access control list (ACL)** directives that are prescriptive—that is, these directives apply to all entries in the subtree below the ACP where this attribute is defined. Any entry in the directory can contain values for this attribute. Access to this attribute itself is controlled in the same way as access to any other attribute.

> **Note:** It is possible to represent ACL directives specific to a single entry in the orclACI attribute. However, in such scenarios, for administrative convenience and performance advantages, Oracle Corporation recommends using orclEntryLevelACI—discussed in "orclEntryLevelACI" on page 13-3. This is because the LDAP operational overhead increases with the number of directives represented through orclACI. You can reduce this overhead by moving entry specific directives from orclACI to orclEntryLevelACI.

orclEntryLevelACI

When a policy pertains only to a specific entity—for example, a special user—you can maintain, within a single entry, the ACL directives specific to that entry. Oracle Internet Directory enables you to do this through a user-modifiable operational attribute called orclEntryLevelACI. The orclEntryLevelACI attribute contains ACL directives that apply to only the entry with which it is associated.

Any directory entry can optionally carry a value for this attribute. This is because Oracle Internet Directory extends the abstract class top to include orclEntryLevelACI as an optional attribute.

The orclEntryLevelACI attribute is multi-valued and has a structure similar to that of orclACI. The structure definition is provided later in this chapter.

Privilege Groups

Group entries in Oracle Internet Directory are associated with either the groupOfNames or the groupOfUniqueNames object class. Membership in the group is specified as a value of the member or uniqueMember attribute respectively.

It is possible to specify access rights for a group of people or entities. Such groups are called privilege groups and are associated with the orclPrivilegeGroup object class.

To grant access rights to a group of users, you create a group entry in the usual way, then associate it with the `orclPrivilegeGroup` object class. You then specify the access policies applicable to that group.

Entries can have either direct memberships to groups, or indirect memberships to other groups by means of nested groups, thus forming a forest of privilege groups. Access policies specified at a given level are applicable to all the members directly or indirectly below it.

Because Oracle Internet Directory evaluates for access control purposes only groups marked as privilege groups, it does not allow setting access policies for non-privilege groups. When a user binds with a specific distinguished name (DN), Oracle Internet Directory computes the user's direct membership in privilege groups. Once it knows the first level groups for the given DN, Oracle Internet Directory computes nesting of all these first level groups into other privilege groups. This process continues until there are no more nested groups to be evaluated.

It is imperative that all groups created for access control purposes, nested or otherwise, be marked as privilege groups by associating them with the `orclPrivilegeGroup` object class. A normal group will not be considered for access control purposes even though it may be a member of a privilege group.

For example, consider the following group of entries, each of which, with the exception of group4, is marked as a privilege group (`objectclass:orclprivilegegroup`). You can set access control policies that apply to the members of group1, group2, and group3.

Group 1

```
dn: cn=group1, c=us
cn: group1
objectclass: top
objectclass: groupofUniquenames
objectclass: orclprivilegegroup
uniquemember:  cn=mary smith,
c=us
uniquemember:  cn=joe smith,
c=us
```

Group 2

```
dn: cn=group2, c=us
cn: group2
objectclass: top
objectclass: groupofUniquenames
objectclass: orclprivilegegroup
uniquemember:  cn=mary jones,
c=us
uniquemember:  cn=joe jones,
c=us
```

Group 3

```
dn: cn=group3, c=us
cn: group3
objectclass: top
objectclass: groupofUniquenames
objectclass: orclprivilegegroup
uniquemember:  cn=group2, c=us
uniquemember:  cn=group1, c=us
uniquemember: cn=group4, c=us
```

Group 4

```
dn: cn=group4, c=us
cn: group4
objectclass: top
objectclass: groupofUniquenames
uniquemember:  cn=john doe, c=uk
uniquemember:  cn=jane doe, c=uk
uniquemember cn=anne smith, c=us
```

Group `cn=group3,c=us` contains the following nested groups:

- `cn=group2,c=us`
- `cn=group1,c=us`
- `cn=group4,c=us`

Access control policies for group3 are applicable to members of group3, group1, and group2 because each of them is marked as a privilege group. These same access control policies are not applicable to the members of group4 because group4 is not marked as a privilege group.

For example, suppose that the user binds to Oracle Internet Directory as a member of group 4 with the DN `cn=john smith,c=uk`. None of the access policies applicable to the members of group3 will apply to this user. This is because his only direct membership is to a non-privilege group. By contrast, if the user were to bind as `cn=john smith,c=us`—that is, as a member of group1 and group2—then his access rights will be governed by access policies set up for members of group1, group2, as well as group3 (in which group1 and group2 are nested). This is because all three groups are associated with the object class `orclPrivilegeGroup`.

Access Control Information Components

Access control information associated with a directory object represents the permissions on the given object that various directory user entities (or subjects) have. Thus, an ACI consists of:

- The object to which you are granting access
- The entities or subjects to whom you are granting access
- The kind of access you are granting

Object: To What Are You Granting Access?

The *object* part of the access control directive determines the entries and attributes to which the access control applies. It can be either an entry or an attribute. Entry objects associated with an ACI are implicitly identified by the entry or the subtree where the ACI itself is defined. Any further qualification of objects at the level of attributes is specified explicitly in the ACL expressions.

In the `orclACI` attribute, the entry DN component of the object of the ACI is implicitly that of all entries within the subtree starting with the ACP as its topmost entry. For example, if `dc=com` is an ACP, then the directory area governed by its ACI is:

```
.*, dc=com.
```

However, since the directory area is implicit, the DN component is neither required nor syntactically allowed.

In the `orclEntryLevelACI` attribute, the entry DN component of the object of the ACL is implicitly that of the entry itself. For example, if `dc=acme,dc=com` has an entry level ACI associated with it, the entry governed by its ACI is exactly: `dc=acme,dc=com`. Since it is implicit, the DN component is neither required nor syntactically allowed.

The object portion of the ACL allows entries to be optionally qualified by a filter matching some attribute(s) in the entry:

```
filter=(ldapFilter)
```

where *ldapFilter* is a string representation of an LDAP search filter. The special entry selector * is used to specify all entries.

Attributes within an entry are included in a policy by including a comma-separated list of attribute names in the object selector.

```
attr=(attribute_list)
```

Attributes within an entry are excluded from a policy by including a comma-separated list of attribute names in the object selector.

```
attr!=(attribute_list)
```

> **Note:** Access to the entry itself must be granted or denied by using the special object keyword ENTRY. Note that giving access to an attribute is not enough; access to the entry itself through the ENTRY keyword is necessary.

> **See Also:** Appendix B, "Using Access Control Directive Format" for information about the format or syntax of ACIs

Subject: To Whom Are You Granting Access?

This section describes the authentication mode, called the bind mode, used to verify the identity of the subject, also called the entity, to whom access is granted.

Bind Mode The bind mode specifies the method of authentication to be used by the subject. There are four modes:

- `Simple`: Simple password-based authentication
- `SSLNoauth`: For SSL-based clients with either anonymous or simple password based authentication. This method uses only the encryption feature of SSL.
- `SSLOneway`: For SSL-based clients with server authentication with either anonymous or password based authentication
- `SSLTwoway`: For SSL-based clients with strong authentication through SSL.

Specifying the bind mode is optional. The directory server verifies that the bind mode of the user is compatible with that of the node with which the user is trying to communicate. The bind mode specified on one node must be compatible with that specified on the node with which it is communicating. For example, if you specify `SSLTwoway` authentication on one node, then the other node must also be configured for this type of authentication.

Entity The entity component identifies the entity or entities being granted access. Note that access is granted to entities, not entries.

Entities can be specified by:

- The special "*" identifier, matching any entry
- The keyword SELF matching the entry protected by the access
- A regular expression matching an entry's distinguished name: `dn=`*`regex`*
- The members of a privilege group object: group=*dn*
- An entry listed in a DN-valued attribute in the entry to which the access applies: `dnattr=(`*`dn-valued_attribute_name`*`)`

For example, the `dnattr` specification is used to give access to a group entry to whomever is listed as the owner of the group entry.

Operations: What Access Are You Granting?

The kind of access granted can be one of the following:

- None
- Compare / nocompare
- Search / nosearch
- Browse / nobrowse
- Read / noread
- Selfwrite / noselfwrite
- Write / nowrite
- Add / noadd
- Delete / nodelete

Note that each access level can be independently granted or denied. The `no`*`xxx`* means *`xxx`* permission is denied.

Note that some access permissions are associated with entries and others with attributes.

Access Level	Description	Type of Object
Compare	Right to perform compare operation on the attribute value	Attributes
Read	Right to read attribute values. Even if read permission is available for an attribute, it cannot be returned unless there is browse permission on the entry itself.	Attributes
Search	Right to use an attribute in a search filter	Attributes
Selfwrite	Right to add oneself to, delete oneself from, or modify one's own entry in a list of DNs group entry attribute. Use this to allow members to maintain themselves on lists. For example, the following command allows people within a group to add or remove only their own DN from the member attribute: `access to attr=(member) by dnattr=(member) (selfwrite)` The `dnattr` selector indicates that the access applies to entities listed in the member attribute. The `selfwrite` access selector indicates that such members can add or delete only their own DN from the attribute.	Attributes
Write	Right to modify / add / delete the attributes of an entry.	Attributes
None	No access rights. The effect of granting no access rights to a subject-object pair is to make the directory appear to the subject as though the object were not present in the directory.	Both entries and attributes
Add	Right to add entries under a target directory entry	Entries
Browse	Permission to return the DNs in the search result. It is equivalent to the list permission in X.500. This permission is also required for a client to use an entry DN as the base DN in an ldapsearch operation.	Entries
Delete	Right to delete the target entry	Entries

The entry level access directives are distinguished by the keyword `ENTRY` in the object component.

Note: By default, for both structural and content access items, everyone is given access to read, search, write, and compare all attributes in an entry, and selfwrite permissions are unspecified. If an entry is unspecified, access is determined at the next highest level in which access is specified.

How ACL Evaluation Works

When processing a request, the access level granted to the requester has to be evaluated for each of the attributes involved in the request. This evaluation is done systematically for each attribute associated with every entry involved in an LDAP operation.

The process of evaluating access to any object (attribute in an entry) involves potentially examining all the ACI directives that are applicable for that object. This is because of the hierarchical nature of ACPs and the inheritance of policies from superior ACPs to subordinate ACPs.

The evaluation starts with examining ACI directives in the entry's entry level ACI, `orclEntryLevelACI`. Until the evaluation is complete, the ACP policies are successively considered, starting with the immediate ACP, followed by the chain of its superior ACPs.

The access evaluation is done for the entry and each of its attributes individually. Oracle Internet Directory evaluates entry level access permissions to see whether the given subject is allowed to perform the given operation.

During ACL evaluation, an attribute is said to be in one of the following states:

State	Description
Resolved with permission	The required access for the attribute has been granted in the ACI.
Resolved with denial	The required access for the attribute has been explicitly denied in the ACI.
Unresolved	No applicable ACI has yet been encountered for the attribute in question.

For all operations except search, the evaluation stops if:

- Access to the entry itself is denied
- Any of the attributes reach the resolved with denial state.

In this case the operation would fail and an error would be returned to the client.

For a search operation, the evaluation continues until all the attributes reach the resolved state. Attributes that are resolved with denial are not returned.

ACL Evaluation Precedence Rules

An LDAP operation requires the BindDN, or subject, of the LDAP session to have certain permissions to the objects affected by the operation—including permissions on the entry itself and on the individual attributes of the entry.

Typically, there could be a hierarchy of access control administration authorities, starting from the root of a naming context down to successive administrative points (or access control policy points). An ACP is any entry which has a defined value for the `orclACI` attribute. Additionally, the access information specific to a single entry can also be represented within the entry itself (`orclEntryLevelACI`).

ACL evaluation involves determining whether a subject has sufficient permissions to perform an LDAP operation. Typically an `orclentryLevelACI` or `orclACI` might not contain all the necessary information for ACL evaluation. Hence, all available ACL information is processed in a certain order until the evaluation is fully resolved.

That order of processing follows these rules:

- The entry level ACI is examined first. ACI in the `orclACI` are examined starting with the ACP closest to the target entry and then its superior ACP and so on.
- At any point, if all the necessary permissions have been determined, the evaluation stops; otherwise, the evaluation continues.
- Within a single ACI, if the entity associated with the session DN matches more than one item identified in the *by* clause, the effective access evaluates to:
  - The union of all the granted permissions in the matching by clause items

    ANDed with
  - The union of all the denied permissions in the matching by clause items

Precedence at the Entry Level ACIs at the entry level are evaluated in the following order:

1. With a filter. For example:

   ```
   access to entry filter=(cn=p*)
       by group1 (browse, add, delete)
   ```

2. Without a filter. For example:

   ```
   access to entry
       by group1 (browse, add, delete)
   ```

Precedence at the Attribute Level At the attribute level, specified ACIs have precedence over unspecified ACIs.

Specified ACIs at the attribute level are evaluated in the following order:

1. Those with a filter. For example:

   ```
   access to attr=(salary) filter=(salary > 10000)
       by group1 (read)
   ```

2. Those without a filter. For example:

   ```
   access to attr=(salary)
       by group1 (search, read)
   ```

Unspecified ACIs at the attribute level are evaluated in the following order:

1. With a filter. For example:

   ```
   access to attr=(*) filter (cn=p*)
       by group1 (read, write)
   ```

2. Without a filter. For example:

   ```
   access to attr=(*)
       by group1 (read, write)
   ```

Assigning More Than One ACI to the Same Object

If there are two or more ACIs at the same ACP for the same object, then only one ACI is checked, and all other ACIs are ignored. For example, suppose you have the following two ACIs at the same ACP for the same entry:

- ACI #1:

  ```
  access to entry
         by dn="cn=admin, dc=us,dc=acme,dc=com" (browse, add, delete)
  ```
- ACI #2:

  ```
  access to entry
         by dn="cn=manager,dc=us,dc=acme,dc=com" (search, read)
  ```

If ACI #2 happens to be checked first, then the access granted specifically to the administrator in ACI #1 is ignored. If an administrator should then seek access to the entry, that access could not be resolved at this level of the hierarchy. The evaluation would have to move progressively up the hierarchy in search of resolution. If no resolution is found, all access is denied.

The solution is to create only one ACI at the same ACP for this entry. For example:

```
access to entry
    by dn="cn=admin, dc=us,dc=acme,dc=com" (browse, add, delete)
    by dn="cn=manager,dc=us,dc=acme,dc=com" (search, read)
```

Similarly, at the attribute level, suppose you have the following two ACIs:

- ACI #1:

  ```
  access to attr=(userpassword)
          by dnattr=(".*,dc=us,dc=acme,dc=com") (none)
  ```
- ACI #2:

  ```
  access to attr=(userpassword)
          by self (read, write)
  ```

If ACI #1 happens to be returned first, it wins, and the access granted to self in ACI #2 is ignored. If a user then wishes to change his or her own password, that access cannot be granted.

As with the ACIs for entries, the solution is to create only one ACI at the same ACP for this attribute. For example:

```
access to attr=(userpassword)
    by dnattr=(".*,dc=us,dc=acme,dc=com) (none)
    by self (read, write)
```

Granting Exclusionary Access to Objects

If an ACI exists for a given object, and you want to specify access to all other objects except that one, then you must verify that the specified objects do not intersect. For example, suppose you have the following two ACIs:

- ACI #1:

  ```
  access to attr=(userpassword)
  by group1 (read, write)
  ```
- ACI #2:

  ```
  access to attr=(*)
  by group2 (read)
  ```

In this case, the two ACIs intersect, that is, both ACIs try to grant access to the `userpassword` attribute, but ACI #2 is unsuccessful. The reason is that, during the evaluation process, ACI #1 wins because, as noted in "ACL Evaluation Precedence Rules" on page 13-11, it is specified. This means that anyone in group2 who tries to access the `userpassword` attribute is not given access at this level of the hierarchy.

The evaluation would have to move progressively up the hierarchy in search of resolution. If no resolution is found, all access is denied.

The solution is to use the following syntax for ACI #1 and ACI #2:

- ACI#1:

  ```
  access to attr=(userpassword)
  by group1 (read, write)by group2 (read)
  ```
- ACI #2:

  ```
  access to attr!=(userpassword)
  by group2 (read)
  ```

In the revised ACI #1, we give to group2 read access to the `userpassword` attribute.

In the revised ACI #2, we negate group2 access to the `userpassword` attribute, and we grant read access to all attributes *except* the `userpassword` attribute.

ACL Evaluation For Groups

If an operation on an attribute or the entry itself is explicitly denied at an ACP low in the DIT, then, typically, the ACL evaluation for the attribute (or entry) is considered "Resolved with Denial." However, if the user of the session (bindDN) is a member of a group object, then the evaluation continues as if it is still unresolved. If permissions are granted to the user of the session at an ACP higher in the tree through a group subject selector, then such grants have higher precedence than any denials lower in the tree.

This scenario is the only case in which ACL policy at a higher level ACP has a higher precedence than that of an ACP lower in the DIT.

Access Level Requirements for LDAP Operations

The following table lists LDAP operations and the access required to perform each one.

Operation	Required Access
Create an object	Add access to the parent entry
Modify	Write access to the attributes that are being modified
ModifyDN	Delete access to the current parent and Add access to the new parent.
ModifyDN (RDN)	Write access to the naming attribute, that is, the RDN attribute

Operation	Required Access
Remove an object	Delete access to the object being removed
Compare	Compare access to the attribute
Search	■ Search access on the filter attributes and browse access on the entry (if only the entry DN needs to be returned as a result) ■ Search access on the filter attributes, browse access on the entry, and read permission on the attributes (for all attributes whose values need to be returned as a result)

Managing Access Control by Using Oracle Directory Manager

You can view and modify access control information configured within ACPs by using either Oracle Directory Manager or command-line tools. This section explains how to accomplish these tasks by using Oracle Directory Manager.

Note: Immediately after installing Oracle Internet Directory, be sure to reset the default security configuration as described in "Task 3: Reset the Default Security Configuration" on page 4-9

This section contains these topics:

- Configuring the Display of ACPs in Oracle Directory Manager
- Configuring Searches for ACPs When Using Oracle Directory Manager
- Viewing an ACP by Using Oracle Directory Manager
- Adding an ACP and Creating Access Items by Using Oracle Directory Manager
- Adding an ACP by Using the ACP Creation Wizard of Oracle Directory Manager
- Modifying ACPs and their ACI Directives by Using Oracle Directory Manager
- Example: Managing ACPs by Using Oracle Directory Manager
- Granting Entry-Level Access by Using Oracle Directory Manager

See Also: Appendix A, "Syntax for LDIF and Command-Line Tools" for a description of command-line tools

Configuring the Display of ACPs in Oracle Directory Manager

Oracle Directory Manager enables you to determine whether the navigator pane displays all ACPs automatically or only as the result of a search. If you have a large number of ACPs, you may want to display them only as the result of a search.

To configure the display of ACPs:

1. In the navigator pane, expand Oracle Internet Directory Servers and select the server you want to configure.
2. On the toolbar, click User Preferences. The User Preferences dialog box appears.
3. Select the Configure Access Control Policy Management tab page.
4. In the Configure Access Control Policy Management tab page, select either:
   - Always display all ACPs
   - Only display ACPs based on search request
5. Click OK.

Note: To effect your changes, you must restart Oracle Directory Manager.

Configuring Searches for ACPs When Using Oracle Directory Manager

For ACP searches, Oracle Directory Manager enables you to specify:

- The root of the search
- The maximum number of entries retrieved
- The time limit of the search
- The search depth

To configure searches for ACP entries:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*, and select Access Control Management.
2. On the toolbar, click Configure ACPs Search. The Configure ACPs Search dialog box appears.
3. In the Root of the Search field, enter the DN of the root of your search, or click Browse to navigate to it.

4. In the Max Results (entries) field, enter the number of entries you want ACP searches to retrieve.
5. In the Max Search Time (seconds) field, enter the maximum number of seconds for the duration of the search.
6. In the Search Depth list, select the level at which you want to search. Options are:
   - One Level: To limit the search to all ACP entries one level down from the root of the search
   - Subtree: To search entries within the entire subtree, including the root of the search
7. Click OK.

Viewing an ACP by Using Oracle Directory Manager

If you configured Oracle Directory Manager always to display ACPs, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then you can locate and view an ACP as follows:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management. All of the defined ACPs appear both below Access Control Management in the navigator pane and in the right pane.
2. In the navigator pane, under Access Control Management, select an ACP to display its information in the right pane.

   You can alternatively double-click an ACP in the right pane to display the data in its own window.

The three fields in the Access Control Management pane are:

Field	Description
Path to the Subtree Access Control Point	Contains the path defined by the ACP. If you have navigated down a tree to this point, the path to this point appears in this field. If you are creating a new ACP, you must enter the path to it here.
Structural Access Items (Entry Level Operations)	Lists access to entries. Items listed in the Structural Access Items box identify an entry by the following categories: - By Whom: To whom or what you are granting access (the subject) - Bind Mode: Whether bind mode (authentication) is used - Access rights: Browse, Add, and Delete **See Also:** "Modifying Structural Access Items of an ACP by Using Oracle Directory Manager" on page 13-40 for instructions on how to modify structural access items
Content Access Items (Attribute Level Operations)	Lists items related to attributes within the entry or entries identified in the Entry Filter column. Columns in this window include: - By Whom: To whom or what you are granting access (the subject) - Bind Mode: Whether bind mode (authentication) is used - Op: The matching operation to be performed against the attribute. Choices are EQ (=) and NEQ (!=) - Attribute: The specific attribute to which access is granted or denied (the object) - Access rights: Read, Search, Write, Selfwrite, or Compare access **See Also:** "Modifying Content Access Items of an ACP by Using Oracle Directory Manager" on page 13-43 for instructions on how to modify content access items.

If you configured Oracle Directory Manager to display ACPs only as the result of a search, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then you can locate and view an ACP as follows:

1. Expand Oracle Internet Directory Servers > *directory_server_instance*, then select Entry Management. Perform a search for the entry designated as an ACP. The search result appears in the Distinguished Name box in the lower half of the right pane.

2. In the Distinguished Name box, double-click the entry. The corresponding Entry dialog box appears.
3. To view subtree access controls for this ACP, select the Subtree Access tab.

   To view entry level access controls for this ACP, select the Local Access tab.

Adding an ACP and Creating Access Items by Using Oracle Directory Manager

1. If you configured Oracle Directory Manager always to display ACPs, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*.

   b. Select Access Control Management, and go to step 2.

   If you configured Oracle Directory Manager to display ACPs only as the result of a search, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management.

   b. Select a node where you want the ACP to reside. If there are no ACPs yet configured, then you may select ACPs under "DSE Root".
2. On the toolbar, click Create. A New Access Control Point dialog box appears.
3. In the Path to Entry field, enter the distinguished name (DN) of the entry that will be the ACP. You can alternatively find the DN by looking in the navigator pane under Entry Management or by clicking Browse.
4. To define structural access items (entries), just below the Structural Access Items window, click Create. The Structural Access Item dialog box appears. It has three tabs: Entry Filter, By Whom, and Access Rights.
5. If appropriate, use the Entry Filters tab page to identify the entries to which you are specifying access. In an ACP, the access rights defined apply to the entry and all its subentries unless other filters restrict access further. If you want all entries below the ACP to be governed by the ACP, then you do not need to enter anything on this tab page; simply proceed to the next step.

   You might restrict access to an entry based on one or more of that entry's attributes. For example, you might choose to restrict access to all entries in which the title is manager and in which the organization unit is Americas.

To identify an entry to which you are specifying access:

a. From the menu at the left end of the bar, select an attribute.

b. From the menu in the middle of the bar, select one of the following filter options:

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.

6. Select the By Whom tab page.

   a. Specify the type of authentication—called bind mode—to be used by the subject (that is, the entity that seeks access). There are five bind modes from which to select:

Bind Mode	Description
None	No authentication
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption / decryption is used.
SSL One Way	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

The bind mode is optional in subject specification. If you do not set an authentication method, or choose None, any kind of authentication is accepted. The bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

b. Specify the entity or entities to whom you are granting access.

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User’s Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User’s Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

7. Select the Access Rights tab page.

a. Specify what kinds of rights are granted:

- Browse—Allows the subject to see the entry
- Add—Allows the subject to add other entries below this entry
- Delete—Allows the subject to delete the entry
- Unspecified—Determines access at the next highest level at which access is specified

b. Click OK.

8. To define content access items (attributes), just below the Content Access Items window, click Create. The Content Access Item dialog box appears. Each tab page contains items you can modify.

9. Specify the items in the Entry Filter tab page (if applicable) as described in Step 5 on page 13-19.

10. Select the By Whom tab page and specify the items as described in Step 6 on page 13-21.

11. Select the Attribute tab page.

    a. From the right menu, select the attribute to which you want to grant or deny access.

    b. From the left menu, select the matching operation to be performed against the attribute. Choices are EQ (Equal (=)) and NEQ (Not Equal (!=)).

       For example, if you select EQ and `cn`, then the access rights you grant apply to the `cn` attribute. If you select NEQ and `cn`, then the access rights you grant do not apply to the `cn` attribute.

12. Select the Access Rights tab page and specify the items as described in Table 13–1.

Table 13–1 Access Rights for Attributes

Access Right	Description
Compare	Right to perform compare operation on the attribute value
Read	Right to read attribute values. Even if read permission is available for an attribute, it cannot be returned unless there is browse permission on the entry itself.
Search	Right to use an attribute in a search filter
Selfwrite	Right to add oneself to, delete oneself from, or modify one's own entry in a list of DNs group entry attribute. Use this to allow members to maintain themselves on lists. For example, the following command allows people within a group to add or remove only their own DN from the member attribute: `access to attr=(member) by dnattr=(member) (selfwrite)` The `dnattr` selector indicates that the access applies to entities listed in the member attribute. The `selfwrite` access selector indicates that such members can add or delete only their own DN from the attribute.
Write	Right to modify / add / delete the attributes of an entry.

13. Click OK to close this dialog box and return to the main Oracle Directory Manager dialog box.

Adding an ACP by Using the ACP Creation Wizard of Oracle Directory Manager

1. If you configured Oracle Directory Manager always to display ACPs, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*.

   b. In the navigator pane, select Access Control Management, and go to step 2.

   If you configured Oracle Directory Manager to display ACPs only as the result of a search, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management.

   b. In the navigator pane, select a node where you want the ACP to reside. If there are no ACPs yet configured, you may select ACPs under "DSE Root".

2. On the toolbar, click Create. A New Access Control Point dialog box appears.

3. In the Path to Entry field, enter the distinguished name (DN) of the entry that will be the ACP. You can alternatively find the DN by looking in the navigator pane under Entry Management or by clicking Browse.

4. To define structural access items (entries), just below the Structural Access Items window, click Create via Wizard. The first Structural Access Item dialog box appears.

In an ACP, the access rights defined apply either to the entry and all its subentries or to a specific entry only. The next sections tell you how to configure an ACP for either option.

Specifying Prescriptive Structural Access Items

If you specify prescriptive structural access items, then all entries below the ACP are governed by that ACP.

If you want prescriptive structural access items, then you do not need to enter anything on this first Structural Access Item dialog box. Follow these steps:

1. Click Next. A second Structural Access Item dialog box prompts you to specify to whom you are granting access.

2. Specify the type of authentication—called bind mode—to be used by the subject (that is, the entity that seeks access). There are five bind modes from which to select:

Bind Mode	Description
None	No authentication
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption / decryption is used.
SSL One Way	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

The bind mode is optional in subject specification. If you do not set an authentication method, or choose None, any kind of authentication is accepted. The bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

3. Specify the entity or entities to whom you are granting access.

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User's Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.

Entity	Description
When Session User’s Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

4. Click Next. A Structural Access Item dialog box prompts you for access rights information. Specify what kinds of rights are granted:
   - Browse: Allows the subject to see the entry
   - Add: Allows the subject to add other entries below this entry
   - Delete: Allows the subject to delete the entry
   - Unspecified: Determines access at the next highest level at which access is specified
5. Click Finish.

Specifying Structural Access Items for a Specific Entry

To specify the entry to which you are assigning structural access, follow these steps:

1. From the menu at the left end of the bar, select an attribute.
2. From the menu in the middle of the bar, select one of the following filter options:

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.

Filter	Description
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

3. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.
4. Click Next. A Structural Access Item dialog box prompts you to specify to whom you are granting access.
   a. Specify the type of authentication—called bind mode—to be used by the subject (that is, the entity that seeks access). There are five bind modes from which to select:

Bind Mode	Description
None	No authentication
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption / decryption is used.
SSL One Way	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

The bind mode is optional in subject specification. If you do not set an authentication method, or choose None, any kind of authentication is accepted. The bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

b. Specify the entity or entities to whom you are granting access.

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User’s Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User’s Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

5. Click Next. A Structural Access Item dialog box prompts you for access rights information. Specify what kinds of rights are granted:

6. Click Finish.

Specifying Prescriptive Content Access Items

If you specify prescriptive content access items, then all entries below the ACP are governed by that ACP.

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server instance*, then select Access Control Management. Just below the Content Access Items window, click Create via Wizard. The first Content Access Item dialog box appears.

   To specify prescriptive content access items, you do not need to enter anything on this first Content Access Item dialog box. Click Next. A second Content Access Item dialog box prompts you to specify to whom you are granting access.

2. Specify the type of authentication—called bind mode—to be used by the subject (that is, the entity that seeks access). There are five bind modes from which to select:

Bind Mode	Description
None	No authentication
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption / decryption is used.
SSL One Way	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

The bind mode is optional in subject specification. If you do not set an authentication method, or choose None, any kind of authentication is accepted. The bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

3. Specify the entity or entities to whom you are granting access.

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User’s Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User’s Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

4. Click Next. A Content Access Item dialog box prompts you to select an attribute and the matching operation to be performed against it.
5. In the Attribute field of the Content Access Item dialog box:
   a. From the right list, select the attribute to which you want to grant or deny access.
   b. From the left list, select the matching operation to be performed against the attribute. Choices are EQ (Equal (=)) and NEQ (Not Equal (!=)).
   c. Click Next. A Content Access Item dialog box prompts you to specify access rights.
6. Specify what kinds of rights are granted as described in Table 13–1 on page 13-23.
7. Click Finish.

Specifying Content Access Items for a Specific Entry

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server instance*, then select Access Control Management.
2. Just below the Content Access Items window, click Create via Wizard. The first Content Access Item dialog box appears.

   To specify the entry to which you are assigning content access, follow these steps:

   a. From the menu at the left end of the bar, select an attribute.
   b. From the menu in the middle of the bar, select one of the following filter options:

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter

Filter	Description
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.

d. Click Next. A second Content Access Item dialog box prompts you to specify to whom you are granting access.

e. Specify the type of authentication—called bind mode—to be used by the subject (that is, the entity that seeks access). There are five bind modes from which to select:

Bind Mode	Description
None	No authentication
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption / decryption is used.
SSL One Way	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

The bind mode is optional in subject specification. If you do not set an authentication method, or choose None, any kind of authentication is accepted. The bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

f. Specify the entity or entities to whom you are granting access.

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User's Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User's Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

g. Click Next.

3. In the Attribute field of the Content Access Item dialog box:

   a. From the right list, select the attribute to which you want to grant or deny access.

   b. From the left list, select the matching operation to be performed against the attribute. Choices are EQ (Equal (=)) and NEQ (Not Equal (!=)).

   c. Click Next.

4. Specify what kinds of rights are granted as described in Table 13–1 on page 13-23.
5. Click Finish.

Modifying ACPs and their ACI Directives by Using Oracle Directory Manager

ACPs are entries that contain prescriptive, that is, inheritable, access control information. This information affects the entry itself and all entries below it. You will most likely create ACPs to broadcast large-scale access control throughout a subtree.

Adding Structural Access Items to an ACP by Using Oracle Directory Manager

1. If you configured Oracle Directory Manager always to display ACPs, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then begin as follows:
   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management. Select Access Control Management. All of the defined Access Control Policy Points (ACPs) appear in a list below Access Control Management in the navigator pane. They also appear in the right pane.
   b. Under Access Control Management, select an ACP to display its information in the right pane, or double-click an ACP in the right pane to display the data in its own dialog box.

If you configured Oracle Directory Manager to display ACPs only as the result of a search, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then begin as follows:

a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management, and select the ACP you want to modify.The information for that ACP is displayed in the right pane.

b. Click Edit. The Subtree Access Control Point dialog box appears.

2. Just below the Structural Access Items box, click Create. The Structural Access Items dialog box displays three tabs: Entry Filter, By Whom, and Access Rights.

3. Use the Entry Filters tab page to narrow the set of entries to which you are granting access. If you want all entries below the ACP to be governed by the ACP, you do not need to use this tab page.

   You might choose an entry based on one or more attributes. For example, you might choose to search for all those whose title is administrative assistant, or for all those whose title is manager and whose organization unit is Americas.

   In the Criteria box of the Entry Filters tab page, use the search criteria bar to select an attribute, enter a value for that attribute, and specify a filter for matching the specified attribute with the value you entered. To do this:

   a. From the menu at the left end of the bar, select an attribute.

   b. From the menu in the middle of the bar, select one of the following filter options:

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.

Filter	Description
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

   c. In the text field at the right end of the search criteria bar, type the value for the attribute you selected.

4. Select the By Whom tab page to define the subject of the ACI.

   a. Specify the type of authentication—called bind mode—to be used by the subject (that is, the entity that seeks access). The bind mode is optional in subject specification. However, for the directive to be applicable, the bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

   There are five bind modes from which to select:

Bind Mode	Description
None	No authentication
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption / decryption is used.
SSL One Way	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

b. Specify the entity or entities to whom you are granting access. Options are:

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User's Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User's Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

c. Click OK.

5. Select the Access Rights tab page.

   a. Select the appropriate options to specify the kinds of rights you want to grant: Browse, Add, or Delete.

   b. Click OK to close the Structural Access Items dialog box and return to the main Oracle Directory Manager window. The structural ACI you just set is listed in the Structural Access Items window of the main Oracle Directory Manager dialog box.

Adding Content Access Items to an ACP by Using Oracle Directory Manager

1. If you configured Oracle Directory Manager always to display ACPs, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management. Select Access Control Management. All of the defined Access Control Policy Points (ACPs) appear in a list below Access Control Management in the navigator pane. They also appear in the right pane.

b. Under Access Control Management, select an ACP to display its information in the right pane, or double-click an ACP in the right pane to display the data in its own dialog box.

If you configured Oracle Directory Manager to display ACPs only as the result of a search, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then begin as follows:

a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management, and select the ACP you want to modify.The information for that ACP is displayed in the right pane.

b. Click Edit. The Subtree Access Control Point dialog box appears.

2. In the Content Access Items window, select the Content Access Item you want to modify.

3. Just below the Content Access Item box, click Create. The Content Access Items dialog box appears.

4. Use the Entry Filters tab page to narrow the set of entries to which you are granting access. If you want all entries below the ACP to be governed by the ACP, you do not need to use this tab page.

   You might choose an entry based on one or more attributes. For example, you might choose to search for all those whose title is administrative assistant, or for all those whose title is manager and whose organization unit is Americas.

   In the Criteria box of the Entry Filters tab page, use the search criteria bar to select an attribute, enter a value for that attribute, and specify a filter for matching the specified attribute with the value you entered. To do this:

   a. From the menu at the left end of the bar, select an attribute.

   b. From the menu in the middle of the bar, select one of the following filter options:

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter

Filter	Description
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

   c. In the text field at the right end of the search criteria bar, type the value for the attribute you selected.

5. Select the By Whom tab page to define the subject of the ACI.

   a. Specify the type of authentication—called bind mode—to be used by the subject (that is, the entity that seeks access). The bind mode is optional in subject specification. However, for the directive to be applicable, the bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

   There are five bind modes from which to select:

Bind Mode	Description
None	No authentication
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption / decryption is used.
SSL One Way	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.

Bind Mode	Description
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

b. Specify the entity or entities to whom you are granting access. Options are:

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User's Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User's Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

c. Click OK.

6. Select the Attribute tab page.

   a. From the right list, select the attribute to which you want to grant or deny access.

   b. From the left list, select the matching operation to be performed against the attribute. Choices are EQ (Equal (=)) and NEQ (Not Equal (!=)).

7. Select the Access Rights tab page.

   a. Select the appropriate options to specify the kinds of rights you want to grant as described in Table 13–1 on page 13-23.

   b. Click OK to close the Structural Access Items dialog box and return to the main Oracle Directory Manager window. The structural ACI you just set is listed in the Structural Access Items window of the main Oracle Directory Manager dialog box.

8. Click OK.

Modifying Structural Access Items of an ACP by Using Oracle Directory Manager

1. If you configured Oracle Directory Manager always to display ACPs, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management. Select Access Control Management. All of the defined Access Control Policy Points (ACPs) appear in a list below Access Control Management in the navigator pane. They also appear in the right pane.

   b. Under Access Control Management, select an ACP to display its information in the right pane, or double-click an ACP in the right pane to display the data in its own dialog box.

   If you configured Oracle Directory Manager to display ACPs only as the result of a search, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management.

   b. Select the ACP you want to modify.The information for that ACP is displayed in the right pane.

2. In the Structural Access Items window, select the item you want to modify, and, just below the Structural Access Items window, click Edit. The Structural Access Item dialog box appears.

3. Use the Entry Filters tab page to narrow the set of entries to which you are granting access. If you want all entries below the ACP to be governed by the ACP, proceed to the next step.

   You might choose an entry based on one or more attributes. For example, you might choose to search for all those whose title is secretary, or for all those whose title is manager and whose organization unit is Americas.

   In the Criteria window of the Entry Filters tab page, use the search criteria bar to select an attribute, enter a value for that attribute, and specify a filter for matching the specified attribute with the value you entered. To do this:

   a. From the menu at the left end of the bar, select an attribute.

b. From the menu in the middle of the bar, select one of the following filter options:

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.

4. Select the By Whom tab page.

a. Specify the type of authentication—called bind mode—to be used by the subject (that is, the entity that seeks access). There are five bind modes from which to select:

Bind Mode	Description
None	No authentication
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption/decryption is used.

Bind Mode	Description
SSL One Way	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

The bind mode is optional in subject specification. For the directive to be applicable, the bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

b. Specify the entity or entities to whom you are granting access.

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User’s Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User’s Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

5. Select the Access Rights tab page.

a. Determine what kinds of rights are granted: Browse, Add, Delete, or Unspecified. If an entry is unspecified, then access is determined at the next highest level in which access is specified.

b. Click OK.

Modifying Content Access Items of an ACP by Using Oracle Directory Manager

1. If you configured Oracle Directory Manager always to display ACPs, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management. Select Access Control Management. All of the defined Access Control Policy Points (ACPs) appear in a list below Access Control Management in the navigator pane. They also appear in the right pane.

   b. Under Access Control Management, select an ACP to display its information in the right pane, or double-click an ACP in the right pane to display the data in its own dialog box.

   If you configured Oracle Directory Manager to display ACPs only as the result of a search, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-16, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* instance > Access Control Management.

   b. Select the ACP you want to modify.The information for that ACP is displayed in the right pane.

2. In the Content Access Items box, select the content access item you want to modify, then, just below the Content Access Item window, click Edit. The Content Access Items dialog box appears. Each tab page contains items you can modify.

3. Specify the items in the Entry Filter tab page (if applicable) as described in "Modifying Structural Access Items of an ACP by Using Oracle Directory Manager" on page 13-40.

4. Select the By Whom tab page and specify the items as described in the section "Modifying Structural Access Items of an ACP by Using Oracle Directory Manager" on page 13-40.

5. Select the Attribute tab page.

   a. From the right menu, select the attribute to which you want to grant or deny access.

   b. From the left menu, select the matching operation to be performed against the attribute. Choices are EQ (Equal (=)) and NEQ (Not Equal (!=)).

6. Select the Access Rights tab page and specify the items as described in the section "Modifying Structural Access Items of an ACP by Using Oracle Directory Manager" on page 13-40.
7. Click OK.

Example: Managing ACPs by Using Oracle Directory Manager

This example illustrates how to use Oracle Directory Manager to create a new ACP that has ACIs within it. Suppose you are an administrator in a large company, and you want to limit access to user passwords, so that everyone can compare a password, but only the owner of each password, that is, the user, can read the password or modify it.

In this example, we create a new ACP and populate it with four ACIs that set the following permissions:

- Limited access to a `userpassword` attribute by everyone
- Open access to the same `userpassword` attribute by the user himself
- Open access to all attributes except `userpassword` to everyone
- Open access to all attributes to everyone

Create a New ACP

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*, and select Access Control Management. A list of ACPs appears in the right pane.
2. Click Create at the bottom of the right pane. A New Access Control Point dialog box appears.
3. In the Path To Entry field, enter the DN where you want the ACP. The ACIs within the ACP will apply to all entries below and including that DN.

Structural Access Items To set the access rights for an entry:

1. Just below the Structural Access Items box, click Create. A Structural Access Items dialog box appears. It contains three tabs: Entry Filter, By Whom, and Access Rights.

   Because you want the ACIs to apply to all entries under the ACP, do not use the Entry Filter tab page.

2. Select the By Whom tab page to define the subject of the ACI. From the Bind Mode list, select the authentication mode appropriate to your environment. To create access rights for everyone, select Everyone. Click OK.
3. Select the Access Rights tab page. By default, all rights—browse, add, and delete—are granted.
   a. Change the access rights so that Everyone can browse all entries, but cannot add or delete them.
   b. Click OK.

Content Access Items The four ACIs in this example use the same structural content item information. They differ only in the content access they allow. The rest of this section describes how to create the content access for the ACIs.

To define the content access items:

1. Below the Content Access Items box, click Create. The Content Access Items dialog box appears.

   Because you want this ACI to apply to all entries under the ACP, do not use the Entry Filter tab page.
2. Select the By Whom tab page, select Everyone, then click OK.
3. Select the Attribute tab page. This page has two fields. The first has two choices: EQ (equals) and NEQ (not equals). The second sets the attribute.

   Select EQ and select `userPassword`.
4. Select the Access Rights tab page. By default, all permissions are granted. Change the permissions so that read, search, write, and compare are denied.
5. Click OK.

   You have completed one ACI.

Create Another ACI

Create another ACI that allows a user to read, write, search, and compare his own password.

1. Under the Content Access Items box, click Create. The Content Access Items dialog box appears.
2. Select the By Whom tab page. Click When Session User’s Distinguished Name (DN) Matches the Accessed Entry, then click OK.

3. Select the Attribute tab page. This tab page has two lists.The first has two choices: EQ (equals) and NEQ (not equals). The second sets the attribute.

   Select EQ and userPassword.

4. Select the Access Rights tab page.

   Grant access to read, search, write, and compare. Leave selfwrite unspecified.

5. Click OK.

You have now created two ACPs. One denies Everyone read, search, write, and compare access to the `userPassword` attribute. The second allows the owner of the password to read, search, write, and compare that attribute.

Create a Third ACI

The next ACI grants access to Everyone to read, search, and compare all attributes except `userPassword`. It denies write access.

1. Under the Content Access Items field, click Create to display the Content Access Items.

2. Select the By Whom tab page.

   Select Everyone, then click OK.

3. Select the Attribute tab page.

   Select NEQ and `userPassword`.

   This combination means that any attribute that is *not* equal to `userpassword` is the object of the permissions in this ACI.

4. Select the Access Rights tab page.

   Grant access to read, search, and compare. Deny write access. Leave selfwrite unspecified.

5. Click OK to apply these permissions and close the dialog box.

Create a Fourth ACI

The next ACI grants access to Self to read, browse, and write all attributes except `userpassword`. Including this ACI avoids any ambiguity about whether Self has the same access permissions as Everyone to attributes other than `userPassword`.

1. Under the Content Access Items field, click Create to display the Content Access Items dialog box.

2. Select the By Whom tab page.

   Click When Session User's Distinguished Name (DN) Matches the Accessed Entry. Click OK.

3. Select the Attribute tab page.

   From the lists, select NEQ and `userPassword`. This combination means that any attribute that is *not* equal to `userPassword` is the object of the permissions in this ACI.

4. Press the Access Rights tab page.

   Grant access to read, search, and write. Leave Selfwrite unspecified.

5. Click OK to apply these permissions and close the dialog box.

Consider other access restrictions you might want to implement. Your directory might contain many entries and attributes that should not be available to everyone.

Granting Entry-Level Access by Using Oracle Directory Manager

To grant entry-level access by using Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Entry Management. You may either:

   - Select the entry to display its properties in the right pane
   - Use the search panel to find the entry, then double-click the entry to open the Entry dialog box.

2. Select the Local Access tab page, then create and edit local ACIs in the Structural Access Item and Content Access Item boxes.

3. Once you have made the changes, click Apply.

> **Note:** You must click Apply to send the information you just entered to the directory server. If you do not click Apply, the information you just entered is simply held in the Oracle Directory Manager cache.

Managing Access Control by Using Command-Line Tools

As described in "Overview of Access Control Policy Administration" on page 13-2, directory access control policy information is represented as user modifiable operational attributes. Hence, you can manage directory access control by using the ldapmodify command to set and alter values of these attributes. Any tool, including ldapmodify and ldapmodifymt, can be used for this purpose.

To directly edit the ACI, you should understand the format and semantics of the directory representation of the ACI. This section contains the formal specification of the ACI format and a description of the semantic issues necessary to manage the ACI using command-line tools.

See Also:

- "LDAP Data Interchange Format (LDIF) Syntax" on page A-2 for information about how to format input by using **LDAP Data Interchange Format (LDIF)**, the required input format for line mode commands
- "ldapmodify Syntax" on page A-15 for information about how to run ldapmodify
- Appendix B, "Using Access Control Directive Format" for information about the format or syntax of ACI

Example: Setting Up an Inheritable ACP by Using ldapmodify

This example sets up subtree access permissions in an `orclACI` at the **root DSE** by using an LDIF file named `my_ldif_file`. Because this example refers to the `orclACI` attribute, this access directive governs all the entries in the DIT.

```
ldapmodify -v -h $1 -D "cn=Directory Manager, o=IMC, c=US" -w "controller" -f
my_ldif_file
```

The LDIF file, `my_ldif_file`, contains the following:

```
dn:
changetype: modify
replace: orclaci
orclaci: access to entry
    by dn="cn=directory manager, o=IMC, c=us" (browse, add, delete)
    by * (browse, noadd, nodelete)
orclaci: access to attr=(*)
    by dn="cn=directory manager, o=IMC, c=us" (search, read, write, compare)
    by self (search, read, write, compare)
    by * (search, read, nowrite, nocompare)
```

Example: Setting Up Entry-Level ACIs by Using ldapmodify

This example sets up entry-level access permissions in the `orclEntryLevelACI` attribute by using an LDIF file named `my_ldif_file`. Because this example refers to the `orclentrylevelACI` attribute, this access directive governs only the entry in which it resides.

```
ldapmodify -v -h myhost -D "cn=Directory Manager, o=IMC, c=US" -w "controller"
-f my_ldif_file
```

The LDIF file, `my_ldif_file`, contains the following:

```
dn:
changetype: modify
replace: orclentrylevelaci
orclentrylevelaci: access to entry
    by dn="cn=directory manager, o=IMC, c=us" (browse, add, delete)
    by * (browse, noadd, nodelete)
orclentrylevelaci: access to attr=(*)
    by dn="cn=directory manager, o=IMC, c=us" (search, read, write, compare)
    by * (search, read, nowrite, nocompare)
```

Note: In this example, no DN value is specified. This means that this ACI pertains to the root DSE and its attributes only.

Example: Using Wild Cards

This example shows the use of wild cards (*) in the object and subject specifiers. For all entries within the `acme.com` domain, it grants to everyone browse permission on all entries, as well as read and search permissions on all attributes.

`orclACI` attribute in the ACP at `dc=com`

```
access to entry by * (browse)
access to attr=(*) by * (search, read)
```

Note that, in order to allow reading the attributes, browse permissions must be granted on the entries in order for read permissions to be granted to the attributes of those entries.

Example: Selecting Entries by DN

This example shows the use of a regular expression to select the entries by DN in two access directives. It grants to everyone read-only access to the address book attributes under dc=acme,dc=com access.

orclACI attribute of dc=acme, dc=com:

```
access to entry by * (browse)
access to attr=(cn, telephone, email) by * (search, read)
```

orclACI attribute of dc=us, dc=acme, dc=com:

```
access to entry by * (browse)
access to attr=(*) by dn=".*,dc=us,dc=acme,dc=com" (search, read)
```

Example: Using Attribute and Subject Selectors

This example shows the use of an attribute selector to grant access to a specific attribute, and various subject selectors. The example applies to entries in the dc=us,dc=acme,dc=com subtree. The policy enforced by this ACI can be described as follows:

- For all entries within the subtree, the administrator has add, delete, and browse permissions. Others within the dc=us subtree can browse, but those outside it have no access to the subtree.
- The salary attribute can be modified by one's manager and viewed by oneself. No one else has access to the salary attribute.
- The userPassword attribute can be viewed and modified by oneself and the administrator. Others can only compare this attribute.
- The homePhone attribute can be read and written by oneself and viewed by anyone else.
- For all other attributes, only the administrator can modify values. Everyone else can compare, search, read, but cannot update attribute values.

"orclACI" attribute of "dc=us, dc=acme, dc=com":

```
access to entry
by dn="cn=admin, dc=us,dc=acme,dc=com" (browse, add, delete)
by dn=".*, dc=us,dc=acme,dc=com" (browse)
by * (none)
```

```
access to attr=(salary)
by dnattr=(manager) (read, write)
by self (read)
by * (none)


access to attr=(userPassword)
by self (search, read, write)
by dn="cn=admin, dc=us,dc=acme,dc=com" (search, read, write)
by * (compare)


access to attr=(homePhone)
by self (search, read, write)
by * (read)


access to attr != (salary, userPassword, homePhone)
by dn="cn=admin, dc=us,dc=acme,dc=com" (compare, search, read, write)
by * (compare, search, read)
```

Example: Granting Read-Only Access

This example gives to everyone read-only access to address book attributes under `dc=acme,dc=com`. It also extends to everyone read access to all attributes within the `dc=us,dc=acme,dc=com` subtree only.

`orclACI` attribute of `dc=acme, dc=com`:

```
access to entry by * (browse)
access to attr=(cn, telephone, email) by * (search, read)
```

`orclACI` attribute of `dc=us, dc=acme, dc=com`:

```
access to entry by * (browse)
access to attr=(*) by dn=".*,dc=us,dc=acme,dc=com" (search, read)
```

Example: Granting Selfwrite Access to Group Entries

This example allows people within the US domain to add or remove only their own name (DN) to or from the member attribute of a particular group entry, for example, a mailing list.

`orclEntryLevelACI` attribute of the group entry in question:

```
access to attr=(member)
by dn=".*, dc=us,dc=acme,dc=com" (selfwrite)
```

Part IV

Directory Replication

This part provides detailed discussions of replication and how to manage it. It contains these chapters:

14

About Directory Replication

In "Distributed Directories" on page 2-22, you saw an overview of replication. This chapter provides a closer look. It contains these topics:

- Directory Replication Groups and Replication Agreements
- Oracle9i Replication
- Replication Architecture
- Change Log Purging
- Conflict Resolution in Replication
- The Replication Process

See Also:

- "Replication" on page 2-22 for a more general, conceptual discussion of replication
- Chapter 15, "Managing Directory Replication" for information on managing replication

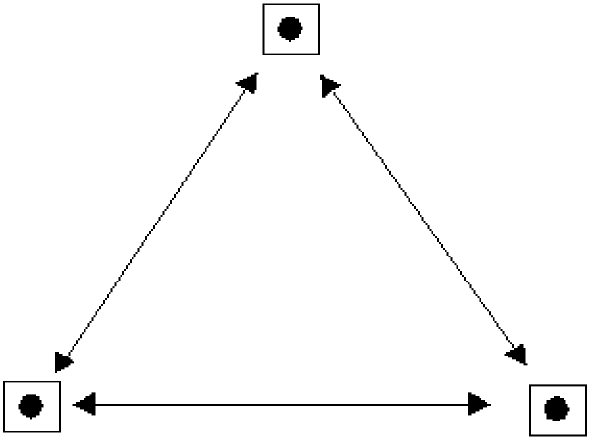
Directory Replication Groups and Replication Agreements

The set of directory servers that participate in replication of a given naming context is called a directory replication group (DRG). A special directory entry, called a replication agreement, represents the replication relationship among the directory servers in a DRG.

It is possible for a directory server to be both a supplier and a consumer of change log information. Oracle Internet Directory uses this feature to support multimaster replication.

Figure 14–1 illustrates a directory replication group in which three nodes share updates with each other in a replication agreement.

Figure 14–1 Directory Replication Group



In Figure 14–1, each bullet represents a node of Oracle Internet Directory. The agreement is identical on each node except for local options such as partitioned naming contexts on the local directory server. The replication agreement on each node lists all the other nodes to which it delivers, and from which it receives, changes.

> **See Also:** "Task 6: Configure Replication" on page 15-10 for information about how to configure replication agreements

Oracle9*i* Replication

Transport of update information between nodes in a replication agreement is managed by Oracle9*i* Replication, a store-and-forward transport feature available in Oracle9*i*. It allows database tables to be kept synchronized across two Oracle databases.

Oracle9*i* Replication stores local changes and periodically propagates them in batches to consumer servers. The consumer replication servers apply the remote changes to the local directory server and then purge the applied remote changes from their local stores.

Oracle9*i* Replication environments allow read and update access to directory tables anywhere in the Oracle9*i* replication group. Typical Oracle9*i* Replication configurations use row-level replication with asynchronous data propagation.

Oracle9*i* Replication provides proven network tolerance and its data transfer can be controlled and monitored by Oracle Enterprise Manager. Such manageability allows a high degree of flexibility in how the data transfer is scheduled.

> **See Also:** *Oracle9i Replication* for information about Oracle9*i* Replication

Replication Architecture

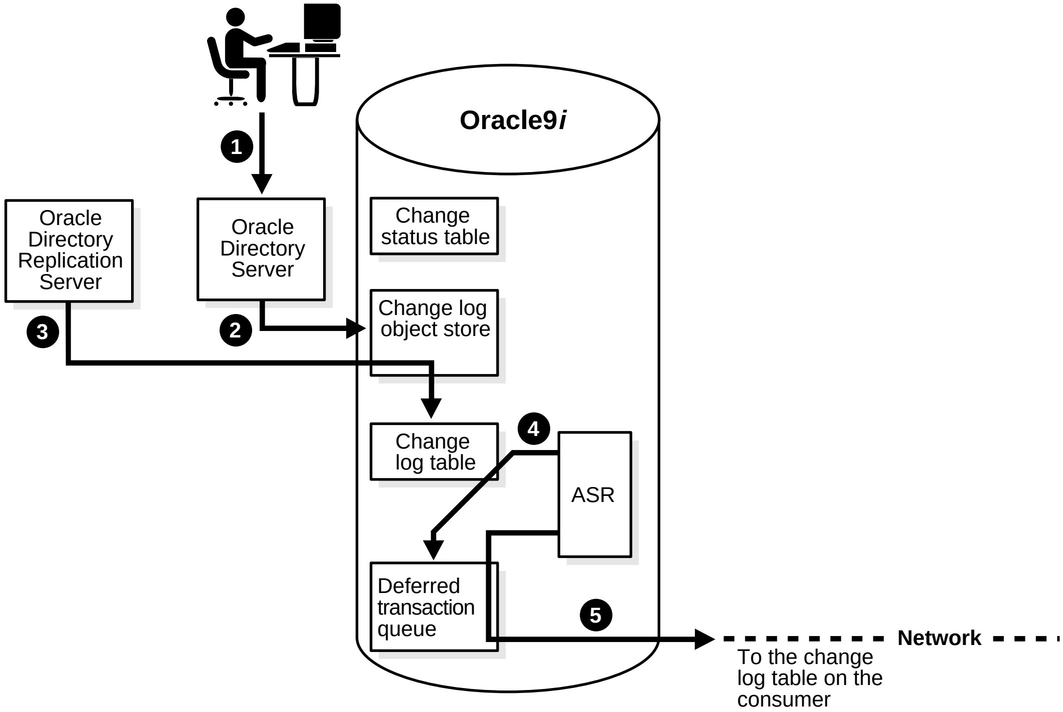
Supplier servers write their changes to change logs, and then regularly send batched directory changes to other supplier and consumer servers. Consumer servers receive the change log data, then reproduce the changes locally.

When you configure replication, you specify which nodes in a replication group share changes. Regardless of the number of nodes you introduce into the replication environment, the basic architecture for replication remains the same. Local changes are distributed to remote nodes and applied by replication server processing. To apply the changes on a remote node, the replication server, acting as a client, sends commands to the directory server that implements them.

The rest of this section discusses, in general terms, the replication process, both from the standpoint of the supplier, and from that of the consumer.

The Replication Process on the Supplier Side

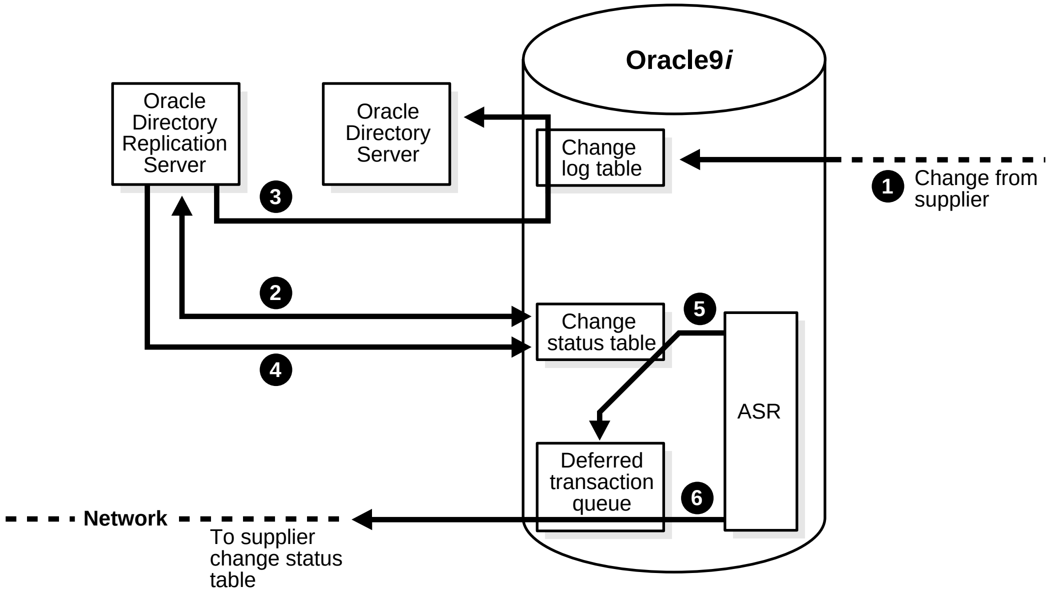
The following graphic and its accompanying text explain what happens on the supplier side during the replication process.



1. An LDAP client issues a directory modification.
2. The Oracle directory server generates a change log object in the change log object store.
3. At a scheduled time, the Oracle directory replication server launches an outbound change log processing thread. This thread translates the change log object into a row—for example, Change entry—in the change log table.
4. When a change entry is committed to the change log table, Oracle9*i* Replication immediately copies the change into the deferred transaction queue.
5. After a scheduled interval, Oracle9*i* Replication pushes pending transactions from the deferred transaction queue across the network to the consumer change log table.

The Replication Process on the Consumer Side

The following graphic and its accompanying text explain the replication process on the consumer side.



1. A change arrives in the consumer change log table from the supplier.
2. The Oracle directory replication server launches a change log processing thread for each supplier, based on a scheduled replication cycle. This thread first consults the change status table for the last change applied from the supplier to the consumer.
3. The Oracle directory replication server then fetches and applies all the new changes from the change log table to the Oracle directory server.
4. The Oracle directory replication server then updates the change status table to record the last change applied from the supplier before exiting.
5. Oracle9*i* Replication copies the change status update into the deferred transaction queue.
6. After the scheduled Oracle9*i* Replication replication interval, Oracle9*i* Replication pushes pending change status updates from the deferred transaction queue to the supplier change status table.

Although, in the previous figures, the roles of supplier and consumer have been separated, in an actual multimaster replication environment, each directory server is both a supplier and a consumer. In such an environment, the purging of entries that are already applied or that have been dropped as candidate changes occurs regularly. Remote change records in the local Changelog table are purged by the garbage collection thread if they have been applied locally. Local change records in the local Changelog table are purged by the garbage collection thread if they have been distributed to all the consumers.

See Also: "Task 6: Configure Replication" on page 15-10 for information on configuring replication

Change Log Purging

Change log purging takes place in Oracle Internet Directory in two ways:

Change number-based

This is the default method. The replication server purges those changes that have already been applied to all the nodes in a DRG.

Time-based

You can run this method to augment change number-based purging. To use this additional method, you set a parameter specifying in hours the lifespan of change log objects. For example, you can set this parameter to purge all change log objects that are 24 hours old. Use this method to prevent the change log from becoming too large.

See Also:

- "Directory Replication Server Parameters" on page 15-11
- "Viewing and Modifying Replication Configuration Parameters by Using Oracle Directory Manager" on page 15-12
- "Modifying Replication Configuration Parameters by Using Command-Line Tools" on page 15-13

Conflict Resolution in Replication

Multimaster replication enables updates to multiple directory servers. Conflicts occur whenever the directory replication server attempts to apply remote changes from a supplier to a consumer and fails for some reason.

The following kinds of LDAP operations can lead to conflicts:

- Addition
- Deletion
- Modification
- Modification of either an RDN or a DN

This section contains these topics:

- Levels at Which Replication Conflicts Occur
- Typical Causes of Conflicts
- Automated Resolution of Conflicts

Levels at Which Replication Conflicts Occur

There are two types of conflicts:

- Entry-level conflicts
- Attribute-level conflicts

Entry-Level Conflicts

An entry-level conflicts is caused when the directory replication server attempts to apply a change to the consumer. Such a change could be one of the following types of changes to the consumer:

- Adding an entry that already exists
- Deleting an entry that does not exist
- Modifying an entry that does not exist
- Applying a modifyrdn operation when the DN does not exist

These conflicts can be difficult to resolve. For instance, it may be impossible to resolve a conflict because:

- The entry has been moved to a different location
- The entry has not yet arrived from a supplier
- The entry has been deleted
- The entry never existed on the consumer

If an entry exists and it should not, then it may be because it was added earlier, or that it recently underwent a modifydn operation.

Attribute-Level Conflicts

An attribute-level conflict is caused when two directories are updating the same attribute with different values at different times. If the attribute is single-valued, then the replication process resolves the conflict by examining the timestamps of the changes involved in the conflict.

Typical Causes of Conflicts

Conflicts usually stem from the timing of changes arising from the occasional slowness or transmission failure over wide-area networks. Also, an earlier inconsistency might continue to cause conflicts if it is not resolved in a timely manner.

Automated Resolution of Conflicts

The directory replication server attempts to resolve all conflicts that it encounters by following this process:

1. The conflict is detected when a change is applied.
2. The replication process attempts to reapply the change a specific number of times or repetitively for a specific amount of time after a specific waiting period.
3. If the replication process reaches the retry limit without successfully applying the change, then it flags the change as a conflict and moves the change to a low-priority, human intervention queue. Changes are then applied according to the time unit specified in the `orclHIQSchedule` parameter in the replication agreement. Before it moves the change, the directory replication server writes the conflict into a log file for the system administrator.

Note: There is no conflict resolution of schema, catalog, and group entries during replication. This is because attempting resolution of such large multi-valued attributes would have a significant negative impact on performance. Be careful to avoid updating such entries from more than one master at a time.

The Replication Process

This section describes how the automated replication process adds, deletes, and modifies entries, and how it modifies DNs and RDNs. It contains these topics:

- How the Replication Process Adds a New Entry to a Consumer
- How the Replication Process Deletes an Entry
- How the Replication Process Modifies an Entry
- How the Replication Process Modifies a Relative Distinguished Name
- How the Replication Process Modifies a Distinguished Name

How the Replication Process Adds a New Entry to a Consumer

When directory replication server successfully adds a new entry to a consumer, it follows this change application process:

1. The directory replication server looks in the consumer for the DN of the parent of the target entry. Specifically, it does this by looking for a **global unique identifier (GUID)** assigned to the DN of the parent.
2. If the parent entry exists, then the directory replication server composes a DN for the new entry and places the new entry under its parent in the consumer. It then places the change entry in the purge queue.

If the change entry is not successfully applied on the first try:

The directory replication server places the new change entry in the retry queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change entry is not successfully applied on *all but the last* **retry:**

The directory replication server keeps the change entry in the retry queue, decrements the number of retries, and repeats the change application process.

If the change entry is not successfully applied on the last retry:

The directory replication server checks to see if the new entry is a duplicate of an existing entry.

If the change entry is a duplicate entry:

The directory replication server applies the following conflict resolution rules:

- The entry with the older creation time stamp is used.
- If both entries have the same creation time stamp, then the entry with the smaller GUID is used.

If the change entry is used, then the target entry is removed, the change is applied, and the change entry is placed in the purge queue.

If the target entry is used, then the change entry is placed in the purge queue.

If the change entry is not a duplicate entry:

The directory replication server places the change entry in the human intervention queue, and repeats the change application process at the interval you specified in the `orclHIQSchedule` parameter.

If the change entry is not successfully applied after it has been placed in the human intervention queue:

The directory replication server keeps the change in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator. The administrator can use the OID reconciliation tool and the human intervention queue manipulation tool to resolve the conflict.

How the Replication Process Deletes an Entry

When the directory replication server deletes an entry from a consumer, it follows this change application process:

1. The directory replication server looks in the consumer for an entry with a GUID matching the one in the change entry.
2. If the matching entry exists in the consumer, then the directory replication server deletes it. It then places the change entry in the purge queue.

If the change entry is not successfully applied on the first try:

The directory replication server places the change entry in the retry queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change entry is not successfully applied on *all but the last* **retry:**

The directory replication server keeps the change entry in the retry queue, decrements the number of retries, and repeats the change application process.

If the change entry is not successfully applied on the last retry:

The directory replication server places the change entry in the human intervention queue and repeats the change application process at specified intervals.

If the change entry is not successfully applied after it has been placed in the human intervention queue:

The directory replication server keeps the change entry in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator. The administrator can use the OID reconciliation tool and the human intervention queue manipulation tool to resolve the conflict.

How the Replication Process Modifies an Entry

When the directory replication server modifies an entry in a consumer, it follows this change application process:

1. The directory replication server looks in the consumer for an entry with a GUID matching the one in the change entry.
2. If the matching entry exists in the consumer, then the directory replication server compares each attribute in the change entry with each attribute in the target entry.
3. The directory replication server then applies the following conflict resolution rules:
    a. The attribute with the most recent modify time is used.
    b. The attribute with the most recent version of the attribute is used—for example, version 1, 2, or 3.
    c. The modified attribute on the host whose name is closest to the beginning of the alphabet is used.

4. The directory replication server applies the filtered modification, and places the change entry in the purge queue.

If the change entry is not successfully applied on the first try:

The directory replication server places the change entry in the retry queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change entry is not successfully applied on *all but the last* **retry:**

The directory replication server keeps the change entry in the retry queue, decrements the number of retries, and repeats the change application process.

If the change entry is *not* **successfully applied by the last retry:**

The directory replication server places the change entry in the human intervention queue and repeats the change application process at specified intervals.

If the change entry is not successfully applied after it has been placed in the human intervention queue:

The directory replication server keeps the change entry in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator. The administrator can use the OID reconciliation tool and the human intervention queue manipulation tool to resolve the conflict.

How the Replication Process Modifies a Relative Distinguished Name

When the directory replication server modifies the RDN of an entry in a consumer, it follows this change application process:

1. The directory replication server looks in the consumer for the DN with a GUID that matches the GUID in the change entry.
2. If the matching entry exists in the consumer, then the directory replication server modifies the RDN of that entry and places the change entry in the purge queue.

If the change entry is not successfully applied on the first try:

The directory replication server places the change entry in the retry queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change entry is not successfully applied on *all but the last* **retry:**

The directory replication server keeps the change entry in the retry queue, decrements the number of retries, and repeats the change application process.

If the change entry is not successfully applied on the last retry:

The directory replication server places the change entry in the human intervention queue and checks to see if it is a duplicate of the target entry.

If the change entry is a duplicate entry:

The directory replication server applies the following conflict resolution rules:

- The entry with the older creation time stamp is used.
- If both entries have the same creation time stamp, then the entry with the smaller GUID is used.

If the change entry is used, then the target entry is removed, the change entry is applied, and then placed in the purge queue.

If the target entry is used, then the change entry is placed in the purge queue.

If the change entry is not a duplicate entry:

The directory replication server places the change entry in the human intervention queue, and repeats the change application process at specified intervals.

If the change entry is not successfully applied after it has been placed in the human intervention queue:

The directory replication server keeps the change entry in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator. The administrator can use the OID reconciliation tool and the human intervention queue manipulation tool to resolve the conflict.

How the Replication Process Modifies a Distinguished Name

When the directory replication server modifies the DN of an entry in a consumer, it follows this change application process:

1. The directory replication server looks in the consumer for the DN with a GUID that matches the GUID in the change entry.

The directory replication server also looks in the consumer for the parent DN with a GUID that matches the GUID of the new parent specified in the change entry.

2. If both the DN and the parent DN of the target entry exist in the consumer, then the directory replication server modifies the DN of that entry and places the change entry in the purge queue.

If the change entry is not successfully applied on the first try:

The directory replication server places the change entry in the retry queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change entry is not successfully applied on *all but the last* **retry:**

The directory replication server keeps the change entry in the retry queue, decrements the number of retries, and repeats the change application process.

If the change entry is *not* **successfully applied by the last retry:**

The directory replication server places the change entry in the human intervention queue and checks to see if it is a duplicate of the target entry.

If the change entry is a duplicate entry:

The directory replication server applies the following conflict resolution rules:

- The entry with the older creation time stamp is used.
- If both entries have the same creation time stamp, then the entry with the smaller GUID is used.

If the change entry is used, then the target entry is removed, the change entry is applied, and then placed in the purge queue.

If the target entry is used, then the change entry is placed in the purge queue.

If the change entry is not a duplicate entry:

The directory replication server places the change entry in the human intervention queue, and repeats the change application process at specified intervals.

If the change entry is not successfully applied after it has been placed in the human intervention queue:

The directory replication server keeps the change entry in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator. The administrator can use the OID reconciliation tool and the human intervention queue manipulation tool to resolve the conflict.

15

Managing Directory Replication

Replication is the mechanism that maintains exact duplicates of specified naming contexts on multiple nodes. This chapter tells you how to install, configure, and manage replication in Oracle Internet Directory.

Note: For release 3.0.1, you can use Oracle Internet Directory replication only if you have installed **Oracle9i Replication**. This ships with all standalone purchases of Oracle Internet Directory and with Oracle9*i* Enterprise Edition. Oracle9*i* Replication is not included with Oracle9*i* Standard Edition.

This chapter contains these topics:

- Installing and Configuring Replication
- Adding a Replication Node
- Deleting a Replication Node
- Resolving Conflicts Manually
- Identifying a Node as Independent of Its Host

See Also: "Replication" on page 2-22 for a conceptual discussion of replication

Installing and Configuring Replication

This section describes how to install and initialize directory replication server software on a node.

Each node in a group of directory servers holds an updatable copy, also called an updatable replica, of the same **naming context** or set of naming contexts. These naming contexts are synchronized with each other by replication processing. This group of nodes is called a **directory replication group (DRG)**.

> **Note:** The instructions in this section apply to setting up replication in a group of empty nodes. For instructions on adding a node to an existing DRG, see "Adding a Replication Node" on page 15-20.
>
> If you are deploying more than one Oracle Internet Directory instance on the same machine, then you cannot uniquely identify each directory server instances by the name of its host. In this case, before installing and configuring replication, follow the instructions in "Identifying a Node as Independent of Its Host" on page 15-32.

To install and configure a replication group, perform these general tasks:

Task 1: Install Oracle Internet Directory on All Nodes in the DRG

Task 2: Decide Which Node Will Serve as the Oracle9i Replication Master Definition Site (MDS)

Task 3: At the MDS, Set Up Oracle9i Replication for a Directory Replication Group

Task 4: Load Data into the Directory

Task 5: Start Oracle Directory Server Instances on All the Nodes

Task 6: Configure Replication

Task 7: Start the Replication Servers on All the Nodes

> **Note:** In Oracle Internet Directory release 3.0.1, procedures and tools are not available to create an environment (directory network) consisting of more than one DRG.

Task 1: Install Oracle Internet Directory on All Nodes in the DRG

Note that the typical installation of the Oracle9*i* Enterprise Edition, which is required for the Oracle Internet Directory, includes **Oracle9i Replication**. By contrast, a typical installation of Oracle9*i* Standard Edition does not include Oracle9*i* Replication.

> **See Also:** Installation documentation for Oracle Internet Directory

Task 2: Decide Which Node Will Serve as the Oracle9*i* Replication Master Definition Site (MDS)

A **master definition site (MDS)** is any of the Oracle Internet Directory databases in which the administrator is going to run the configuration scripts. A remote master site is any site other than the Master Definition Site that participates in Oracle9*i* Replication replication.

You must be able to use **Oracle Net Services** to connect to the MDS database and all other nodes that constitute the DRG.

Task 3: At the MDS, Set Up Oracle9*i* Replication for a Directory Replication Group

The following sections lead you through installing and configuring Oracle9*i* Replication through Oracle Internet Directory installation scripts. More advanced Oracle9*i* Replication users may prefer to configure Oracle9*i* Replication through the Oracle9*i* Replication Manager Tool.

> **See Also:** *Oracle9i Replication* and the online help for Oracle9*i* Replication Manager for information on configuring Oracle9*i* Replication by using the Oracle9*i* Replication Manager

Setting up the Oracle9*i* Replication environment to establish a directory replication group (DRG) requires you to:

- Prepare the Oracle Net Services environment for replication
- Configure Oracle9*i* Replication for directory replication.

Prepare the Oracle Net Services Environment for Replication

Follow these steps, described more fully below, on *all nodes* in the directory replication group to prepare the Oracle Net Services environment:

1. Configure sqlnet.ora.
2. Configure tnsnames.ora.
3. Create rollback table space and rollback segments.
4. Modify the parameters in the initialization parameter file, init.ora.
5. Stop and restart the listener.
6. Stop and restart the Oracle Internet Directory database.

To prepare the Oracle Net Services environment for replication:

1. Configure `sqlnet.ora`.

   The `sqlnet.ora` file should contain the following parameters at minimum:

   ```
   names.directory_path = (TNSNAMES)
   names.default_domain = domain
   ```

   On UNIX, this file is in `$ORACLE_HOME/network/admin`

   On Windows NT, this file is in `ORACLE_HOME\network\admin`

2. Configure `tnsnames.ora`.

   The `tnsnames.ora` file must contain **connect descriptor** information in the following format for all Oracle Internet Directory databases:

   ```
   net_service_name =
       (DESCRIPTION =
         (ADDRESS =
            (PROTOCOL = TCP)
            (HOST = HOST_NAME_OR_IP_ADDRESS)
            (PORT = 1521))
         (CONNECT_DATA =
            (service_name = service_name)))
   ```

   On UNIX, this file is in `$ORACLE_HOME/network/admin`

   On Windows NT, this file is in `ORACLE_HOME\network\admin`

Note: You may domain-qualify the net service name (for example, `sales.com`). Regardless of your choice, be sure that the domain component matches the one specified in the NAMES.DEFAULT_DOMAIN parameter in the `sqlnet.ora` file.

3. Create rollback table space and rollback segments.

   You may want to create multiple rollback segments. You can increase the size of the table spaces and segments to meet your system requirements.

   a. Create a tablespace for rollback segments.

      Execute SQL*Plus by typing the following command:

      ```
      sqlplus system/system_password@net_service_name
      ```

      At the SQL*Plus prompt, type:

      ```
      CREATE TABLESPACE table_space_name
      datafile file_name_with_full_path SIZE 50M REUSE AUTOEXTEND ON NEXT
      10M MAXSIZE max_bulk_update transaction_size ex:500M;
      ```

   b. Create rollback segments.

      At the SQL*Plus prompt, type the following lines for each rollback segment:

      ```
      CREATE ROLLBACK SEGMENT rollback_segment_name
      tablespace table_space_name storage (INITIAL 1M NEXT 1M OPTIMAL 2M
      MAXEXTENTS UNLIMITED);
      ```

      Repeat the `CREATE ROLLBACK SEGMENT` command for each rollback segment entered in the initialization parameter file.

4. Modify the parameters in the initialization parameter file, `init.ora`.

   Type the following lines in the initialization parameter file:

   ```
   rollback_segments = (rollback_segment_name_1, rollback_segment_name_2 ...)
   JOB_QUEUE_PROCESSES = a_minimum_of_total_number_of_LDAP_nodes_minus_one
   SHARED_POOL_SIZE = 20000000
   OPEN_LINKS = a_minimum_of_total_number_of_LDAP_nodes_minus_one
   ```

Note: When setting the number of job queue processes, consider using a number high enough to accommodate any nodes you may want to add in the future.

Ensure that the total **System Global Area (SGA)** does not exceed 50% of your system's physical memory.

> **Note:** Every time a database is started, a System Global Area (SGA) is allocated and Oracle background processes are started. The SGA is an area of memory used for database information shared by the database users. The combination of the background processes and memory buffers is called an Oracle instance.

5. Stop and restart the listener.

   To stop the listener for the Oracle Internet Directory database, use the listener control utility (lsnrctl). Type the following command at the LSNRCTL command prompt:

   ```
   SET PASSWORD password
   STOP [listener_name]
   ```

   `SET PASSWORD` is required only if the password is set in the `listener.ora` file. The password defaults to ORACLE. The default listener name is LISTENER.

   To restart the listener for the Oracle Internet Directory database, type the following command at the LSNRCTL command prompt:

   ```
   START [listener_name]
   ```

6. Stop and restart the Oracle Internet Directory database.

   To stop and restart the Oracle Internet Directory database, you can use SQL*Plus.

   See Also:

   - *Oracle Net Services Administrator's Guide*
   - *Oracle9i Database Administrator's Guide for* instructions on stopping and restarting the database

Configure Oracle9*i* Replication For Directory Replication

To configure Oracle9*i* Replication for the replication group, complete the following steps *from the MDS*:

1. Log on as the Oracle Internet Directory software owner account from a UNIX prompt.
2. Change to the following directory:
   - On UNIX: `$`*`ORACLE_HOME/`*`ldap/bin`
   - On Windows NT: *`ORACLE_HOME`*`\ldap\bin`

   > **Note:** Before proceeding to the next step, connect as the system user on all nodes, including the MDS, from the MDS console. Ensure the following:
   >
   > - The Oracle Internet Directory database is up and running
   > - The Oracle Internet Directory listener is up and running
   > - The connect descriptor is correct
   > - The system password is correct

3. Run the following script from the MDS:

   ```
   ldaprepl.sh -asrsetup
   ```

   This script executes a number of operations.

   - It configures the MDS.
   - It configures the remote master sites.
   - It configures replication push jobs at all sites.
   - It resumes replication at the MDS.
   - It verifies that all steps have completed successfully.

   As the script runs, it asks for the information in the following table, first for the MDS, then for the master sites.

Information	Definition
Host name	Name of the computer
Global name	Net service name of the MDS database, as listed in the file `tnsnames.ora`
System password	system password

After you have provided the necessary information for the first master site, the script asks if there is another master site.

4. Enter `Y` or `N`. If you enter `N`, to indicate that you have identified all sites, then it shows a table of the information you have provided, and asks for confirmation. If it is not correct, then press `N`. The script will start again at the beginning, asking about the MDS again.

   After you have provided all the information, the script asks you to verify the correctness of the information. If the information is correct and you press `Y`, then the script begins configuring the sites.

   This process may take a long time, depending on your system resources and the number of nodes in your DRG. The script keeps you informed of its progress.

Note: If you must interrupt the process before it is complete, then you must start at the beginning. Interrupting the process will not negatively affect your re-installation.

Troubleshooting Tip: If the process fails, then do the following:

1. Check the `$ORACLE_HOME/ldap/admin/logs/ldaprepl.log` file to see the status.
2. Go to the directory `$ORACLE_HOME/ldap/admin` and check the status of replication jobs by running the following command:

```
sqlplus system/password@net_service_name @ldaplogq.sql
```

Run this command for each node in the DRG. Issuing this command should result in no rows being selected. If rows are selected containing the failed status and error messages, then this means that Oracle9*i* Replication set up failed. In this case, you may:

- Run the script from the beginning
- Consult the troubleshooting chapter in *Oracle9i Replication*
- Determine a solution from error message information by consulting an expert in Oracle9*i* Replication

Note: If you have large initial data requirements, then use the bulkload tool to load initial data on all the nodes in the DRG. You must stop the server before using bulkload, and bring it up again afterwards.

See Also:

- *Oracle9i Database Administrator's Guide* for instructions on ensuring that the database and listener are running
- *Oracle Net Services Administrator's Guide* for instructions on ensuring that the connect string is correct
- "bulkload Syntax" on page A-28 for bulkload syntax and usage notes

Task 4: Load Data into the Directory

To do this, follow the instructions in "Managing Entries by Using Bulk Tools" on page 8-18.

Task 5: Start Oracle Directory Server Instances on All the Nodes

To start Oracle directory server instances on all nodes, run the following command:

```
oidctl connect=net_service_name server=oidldapd instance=instance_number_of_
ldap_server flags='-p port' start
```

Note: The `instance_number_of_ldap_server` need not be unique across the entire DRG. For example, you can have `instance=1` on both node A on node B.

See Also: Chapter 6, "Managing the Oracle Directory Server" for more information on starting an Oracle directory server **instance**

Task 6: Configure Replication

You need to configure parameters for:

Directory replication server

Directory replication server configuration parameters are stored as special attributes in directory entries. You can configure replication parameters and replication agreements the same way you configure the Oracle Internet Directory. You can do either of the following:

- View and modify the agreements by using Oracle Directory Manager
- Alter the contents of the configuration entries and agreement entries through the command-line tools, such as ldapadd and ldapmodify

This section explains both approaches.

Replication agreements

Replication agreements are entries that list the member nodes within a replication group that share their changes. Replication agreements are referenced by directory replication server configuration parameters that load when the directory replication server runs.

Important: When you install and configure replication for the first time, you must inform the directory replication server about the existence of the member nodes in the replication agreement. To do this, modify the `orclDirReplGroupDSAs` attribute in the replication agreement. See "Replication Agreement Parameters" on page 15-15 for more information.

Location of Directory Replication Server Configuration Parameters

The directory replication server configuration parameters are stored in the replication server **configuration set entry**, which has the following DN:

```
cn=configset0,cn=osdrepld,cn=subconfigsubentry
```

This entry contains replication attributes that control replication processing. You can modify some of these attributes. Note that the `orclDirReplGroupAgreement` attribute contains a replication agreement identifier. In this release, only one replication agreement is possible.

Directory Replication Server Parameters

The next table lists and describes the directory replication server configuration parameters.

Parameter name	Description	Default Values	Modifiable?
modifyTimestamp	Time of entry creation or modification		No
modifiersName	Name of person creating or modifying the entry		No
orclChangeRetryCount	Single-valued attribute. The number of processing retry attempts for a change-entry before being moved to the human intervention queue. The value for this parameter must be equal to or greater than 1 (one).	10	Yes

Parameter name	Description	Default Values	Modifiable?
orclPurgeSchedule	Single-valued attribute. Specifies purge (garbage collection) interval in minutes. Removes entries that are already applied or have been dropped as candidate changes. This thread is initiated periodically based on the frequency that you set. The value for this parameter must be equal to or greater than 1 (one).	10 minutes	Yes
orclThreadsPerSupplier	Number of worker threads directory replication server provides for each supplier for change log processing. The value for this parameter must be equal to or greater than 1 (one).	5	Yes
orclDirReplGroupAgreement	Multi-valued attribute. Identifies the symmetrical replication agreements for which this server is responsible.	orclagreementid=000001, cn=orclreplagreements	No
orclChangeLogLife	Single-valued attribute. Specifies in hours the time for the life of entries in the change log store. 0 (zero) indicates that this is a change number-based purge. **See Also**: "Change Log Purging" on page 14-6	0	Yes

Viewing and Modifying Replication Configuration Parameters by Using Oracle Directory Manager

To view and modify replication configuration parameters:

1. In the navigator pane, expand Oracle Internet Directory > *directory_server_instance* > Server Management > Replication Server.
2. Select the replication configuration set whose parameters you want to view or modify. The corresponding tab pages appear in the right pane.

   Configuration parameters appear in the General tab page. Use this tab page to view replication configuration parameters, and modify many of them. The following table describes the fields in this tab page.

Field	Description
Modify Timestamp	Time of entry creation or modification in **UTC (Coordinated Universal Time)**. You cannot modify this parameter.
Modifier’s Name	Name of person creating or modifying the entry. You cannot modify this parameter.
Change Retry Count	Type the number of attempts that the conflict resolution process tries to apply each update before giving up and logging the incident. The default is 10.
Purge Schedule	Type the number of minutes in between garbage collections. The replication garbage collection thread removes entries that are already applied or have been dropped as candidate changes. The default is 10.
Number of Threads Per Supplier	Type the number of worker threads the directory replication server provides for each supplier for change log processing. The default is 5.
Set	Type the configuration identifier.
Change Log Life	Type the number of hours for the life of the change log objects. **See Also:** "Change Log Purging" on page 14-6

Modifying Replication Configuration Parameters by Using Command-Line Tools

To modify replication configuration parameters by using command-line tools, use the syntax documented in "ldapmodify Syntax" on page A-15.

Modifying the Garbage Collection Interval by Using ldapmodify This example uses an input file named `mod.ldif` to change the garbage collection interval from the default of 10 minutes to 30 minutes.

1. Edit `mod.ldif` as follows:

```
dn: cn=configset0,cn=osdrepld,cn=subconfigsubentry
changetype: modify
replace: orclPurgeSchedule
orclPurgeSchedule: 30
```

2. Use ldapmodify to update the replication server `configset0` parameter value as follows:

```
ldapmodify -h my_host -p 389 -f mod.ldif
```

3. Restart the directory replication server.

Modifying the Change Log Life Parameter by Using ldapmodify This example uses an input file named mod.ldif to change the change log life parameter to 10 hours:

1. Edit mod.ldif as follows:

   ```
   dn: cn=configset0,cn=oidrepld,cn=subconfigsubentry
   changetype: modify
   replace: orclChangeLogLife
   orclChangeLogLife: 10
   ```

2. Use ldapmodify to update the replication server configset0 parameter value as follows:

   ```
   ldapmodify -h my_host -p 389 -f mod.ldif
   ```

3. Restart the directory replication server.

Modifying the Number of Retries Before a Change Is Moved into the Purge Queue by Using ldapmodify This example uses an input file named mod.ldif to change the number of retry attempts from the default of ten times to five times. Specifically, after attempting to apply an update five times, the update is dropped and logged in the replication log.

1. Edit mod.ldif as follows:

   ```
   dn: cn=configset0,cn=osdrepld,cn=subconfigsubentry
   changetype: modify
   replace: orclChangeRetryCount
   orclChangeRetryCount: 5
   ```

2. Use ldapmodify to update the replication server configset0 parameter value as follows:

   ```
   ldapmodify -h my_host -p 389 -f mod.ldif
   ```

3. Restart the directory replication server.

Modifying the Number of Worker Threads Used in Change Log Processing by Using ldapmodify This example uses an input file named mod.ldif to change the number of worker threads used in change log processing to 7:

1. Edit mod.ldif as follows:

   ```
   dn: cn=configset0,cn=osdrepld,cn=subconfigsubentry
   changetype: modify
   replace: orclthreadspersupplier
   orclthreadspersupplier: 7
   ```

2. Use ldapmodify to update the replication server `configset0` parameter value as follows:

   ```
   ldapmodify -h my_host -p 389 -f mod.ldif
   ```

3. Restart the directory replication server.

   See Also: "Restarting Directory Server Instances" on page 4-7 for instructions on restarting the directory replication server

Replication Agreement Parameters

In the parameter `DirectoryReplicationGroupDSAs`, type all of the host names of the DSAs in the DRG. Be sure that this information is identical on all the nodes.

See Also:

- "Viewing and Modifying Replication Agreement Parameters by Using Oracle Directory Manager" on page 15-16
- "Modifying Replication Agreement Parameters by Using ldapmodify" on page 15-17

Location of Replication Agreement Parameters

Replication agreement parameters are stored in the replication agreement entries which have the following DN:

```
orclAgreementID=id number,cn=orclreplagreements
```

This entry contains attributes that pertain only to the nodes participating in this agreement. You can create multiple replication agreements to manage replication between reciprocating nodes, but you can reference only one of them in your start-server message by using Oracle Directory Manager. For Oracle Internet Directory release 3.0.1, only one replication agreement can be used.

Note: Before you modify replication agreement parameters, be sure that you have started the Oracle Internet Directory on all nodes.

Viewing and Modifying Replication Agreement Parameters by Using Oracle Directory Manager

To view and modify replication agreement parameters by using Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Server Management > Replication Server, and select Default Configuration Set.
2. In the right pane, select the Agreement tab to display the replication agreement.

   The fields in this tab page are described in the following table. You can view the parameters and modify some of them by double-clicking the attributes.

Field	Description	Default Values	Modifiable?
Agreements ID	Unique identifier for a replication agreement.	000001	No
Excluded Naming Contexts	Multivalued attribute. Specifies naming contexts excluded from this replication agreement. Changes to entries in these naming contexts sent from other replicas are not applied on the local node.	None	Yes
Replication Group Nodes	Multi-valued attribute. Specifies nodes participating in symmetrical replication agreement. *Nodes that you specify here share updates with one another.*		Yes
Update Schedule	Replication update interval for new changes and those being retried. The value is in minutes.	1	Yes
Orcl HIQSchedule	Replication update interval for the human intervention queue. The value is in minutes. The value is typically higher that orclUpdateSchedule. This gives administrators time to change the DIT structures when retrying an update fails to resolve a conflict.	10	Yes
Replication Protocol	Specifies the replication protocol used in this replication agreement. The supported protocol is Oracle9*i* Replication.	ODS_ASR_1.0	No

3. If you want to return to the values that appeared when you first opened this pane, then click Revert. If you are satisfied with your changes, then click Apply.

Modifying Replication Agreement Parameters by Using ldapmodify

The following table lists and describes the replication agreement parameters.

Parameter	Description	Default Values	Modifiable?
`orclAgreementID`	Unique identifier for a replication agreement.	000001	No
`orclExcludedNamingcontexts`	Multi-valued attribute. Specifies naming contexts excluded from this replication agreement. Changes to entries in these naming contexts sent from other replicas are not applied on the local node.	None	Yes
`orclDirReplGroupDSAs`	Multi-valued attribute. Specifies nodes participating in symmetrical replication agreement. *Nodes that you specify here share updates with one another.*		Yes
`orclUpdateSchedule`	Replication update interval for new changes and those being retried. The value is in minutes.	1	Yes

Parameter	Description	Default Values	Modifiable?
`OrclHIQSchedule`	Replication update interval for the human intervention queue. The value is in minutes. The value is typically higher that orclUpdateSchedule. This gives administrators time to change the DIT structures when retrying an update fails to resolve a conflict.	10	Yes
`orclReplicationProtocol`	Specifies the replication protocol used in this replication agreement. The supported protocol is Oracle9*i* Replication.	ODS_ ASR_1.0	No

To add more nodes to the values in a replication agreement entry, run ldapmodify at the command line, referencing an LDIF-formatted file.

This example uses an input file named `mod.ldif` to add two nodes to a replication agreement:

1. Edit `mod.ldif` as follows:

   ```
   dn: orclagreementid=000001,cn=orclreplagreements
   changetype: modify
   add: orcldirreplgroupdsas
   orcldirreplgroupdsas: hollis
   orcldirreplgroupdsas: eastsun-11
   ```

2. Use ldapmodify to update the replication server `configset0` parameter value as follows:

   ```
   ldapmodify -h host -p port -f mod.ldif
   ```

3. Restart the directory replication server.

This procedure modifies the entry containing the replication agreement whose DN is `orclagreementid=000001,cn=orclreplagreements`. The input file adds the two nodes, hollis and eastsun-11, into the replication group governed by `oraclagreementid 000001`.

Note: You must include the new nodes—for example, hollis and eastsun-11 in the above sample LDIF file—in the `orclDirReplGroupDSAs` parameter on each node in the replicated environment before you start the replication process.

"Adding a Replication Node" on page 15-20 explains the process of adding a new node to a replication environment.

Because Oracle Internet Directory release 3.0.1 supports only one configuration set for directory replication server, you do not need to specify a configuration set.

Task 7: Start the Replication Servers on All the Nodes

To start replication servers on all nodes, type the following command:

```
oidctl connect=db_connection_string server=oidrepld instance=1
   flags='-h host -p port' start
```

Note that the instance number does not need to be unique across the entire DRG.

See Also: Chapter 6, "Managing the Oracle Directory Server" for information on starting the replication servers

Using the Change Log Flag

You can turn off change logging, which occurs in the Oracle directory server, by using the default value of the `-l` flag in the OID Control Utility command for Oracle directory server from *true* to *false*. This is useful if you suspect that the change log file might not be emptying. However, turning change logging off on a given node means that updates on that node cannot be replicated to other nodes in the DRG.

Using the Multimaster Flag

You can turn off the multimaster flag, which occurs in the directory replication server, by using the default value of the `-m` flag in the OID Control Utility command for Oracle directory server from *true* to *false*. This is useful for reducing performance overhead if you are deploying a single master with read-only replica consumers. The multimaster option controls conflict resolution, which serves no purpose if you are deploying a single master.

See Also: "Conflict Resolution in Replication" on page 14-7

Adding a Replication Node

There are two ways to add a new node to a live replication group.

- Using ldifwrite

  This method, described in this section, is the easier of the two. The process can be fully automated, and the generated file can be used for partial replication. Use this procedure unless your directory is very large. Backup using this method can take up to seven hours for a directory with one million entries.

- Using cold backup

  This method, described in Chapter 16, "Adding a Node to a DRG by Using the Database Copy Procedure", cannot be fully automated and cannot be reused for partial replication. However, cold backup takes much less time for a large directory server. For example, if your directory has more than a million entries, then use this method.

> **Note:** Before you add a replication node, prepare the Oracle Net Services environment. For instructions, see "Prepare the Oracle Net Services Environment for Replication" on page 15-4.

To add a replication node to a functioning DRG of any significant size, follow these steps, each of which is more fully described later in this chapter.

Task 1: Stop the Directory Replication Server on All Nodes

Task 2: Configure the New Node into the LDAP Replication Group on All the Existing Nodes

Task 3: Identify a Sponsor Node and Switch the Sponsor Node to Read-Only Mode

Task 4: Backup the Sponsor Node by Using ldifwrite

Task 5: Perform Oracle9i Replication Add Node Setup

Task 6: Switch the Sponsor Node to Updatable Mode

Task 7: Start the Directory Replication Server on All Nodes Except the New Node

Task 8: Load Data into the New Node by Using bulkload

Task 9: Start LDAP Server on the New Node

Task 10: Configure the LDAP Replication Agreement on the New Node

Task 11: Start the Directory Replication Server on the New Node

Note: Commands shown in the following steps require that the following types of items be stored in the corresponding directories:

- Binaries: `$`*`ORACLE_HOME`*`/bin`
- SQL scripts: `$`*`ORACLE_HOME`*`/ldap/admin`
- UNIX scripts: `$`*`ORACLE_HOME`*`/ldap/bin`

Before beginning Task 1, be sure that all three of these types of items are in the path.

Task 1: Stop the Directory Replication Server on All Nodes

To stop the directory replication server, run the following command on each node in the LDAP replication group:

```
oidctl connect=db_connect_string server=oidrepld instance=1 stop
```

Note: The instance number may not be 1. Check the running process to discover the instance number in use here.

Task 2: Configure the New Node into the LDAP Replication Group on All the Existing Nodes

The following example creates an LDIF file, `add_node.ldif`, and configures it into the replication group on all the existing nodes.

```
dn: orclagreementid=000001,cn=orclreplagreements
changetype: modify
replace: orcldirreplgroupdsas
orcldirreplgroupdsas: host_of_the_new_node
orcldirreplgroupdsas: host_of_existing_node_1
orcldirreplgroupdsas: host_of_existing_node_2
.
.
.
orcldirreplgroupdsas: host_name_of_existing_node_n
```

Run the following command against each node in the LDAP replication group:

```
ldapmodify -h host_name_of_the_node -p port -f add_node.ldif
```

Note: This command can be run from one work station for all nodes.

Task 3: Identify a Sponsor Node and Switch the Sponsor Node to Read-Only Mode

A sponsor node is one that will supply the data to the new node. To identify a sponsor node and switch it to read-only mode:

1. Create a new file, `change_mode.ldif`, containing the following:

   ```
   dn:
   changetype: modify
   replace: orclservermode
   orclservermode: r
   ```

2. Run the following commands against the identified sponsor node:

   ```
   ldapmodify -D "cn=orcladmin" -w welcome -h host_name_of_sponsor_node
   -p port -f change_mode.ldif

   oidctl connect=net_service_name server=oidldapd restart
   ```

This restarts all running Oracle directory servers on the sponsor node in Read-Only mode. It takes approximately fifteen seconds for a directory server to restart.

Note: While the sponsor node is in read-only mode, you may not make any updates to it. You may, however, update any of the other nodes, but those updates are not replicated immediately.

Also, the sponsor node and the **MDS** may be the same node.

Task 4: Backup the Sponsor Node by Using ldifwrite

Because this may take a long time, you may start "Task 5: Perform Oracle9i Replication Add Node Setup" while backup is in process.

Enter the following command:

```
ldifwrite -c db_connect_string -b "" -f output_ldif_file
```

Task 5: Perform Oracle9*i* Replication Add Node Setup

You can perform this task at the same time as you are performing "Task 4: Backup the Sponsor Node by Using ldifwrite".

From the sponsor node, run the following script:

```
ldaprepl.sh -addnode
```

This script executes a number of operations.

- It quiesces Oracle9*i* Replication at the sponsor node and any other existing **master site**.
- It configures the master sites and the new node. A master site is any site other than the sponsor node that participates in LDAP replication.
- It configures replication push jobs at all sites including the new node.
- It checks that all steps have completed successfully. (This may take a long time.)
- It performs post-add-node operation.

As the script runs, it asks for the information in Table 15–1, first for the sponsor node then for the existing master sites.

Table 15–1 Oracle9*i* Replication ***Setup Information***

Information	**Description**
Host Name of sponsor node	Name of the computer
Global name	Net service name of the MDS or master site database, as listed in `tnsnames.ora`
system password	system password

When you have identified all the existing master sites, enter `N`. The script then asks for information regarding the new node. Once you have provided that information, the script shows you a table of the information you have provided, and asks for confirmation.

If the information is not correct, then press `N`. The script then starts again at the beginning, asking the same information. If the information is correct and you enter `Y`, then the script begins configuring the sites.

This process can take a long time, depending on your system resources and the size of your DRG. The script keeps you informed of its progress.

Note: If for any reason you must interrupt the process before it is complete, then you must start from the beginning.

Troubleshooting Tip: If the process fails, then do the following:

1. Check the `$ORACLE_HOME/ldap/admin/logs/ldaprepl.log` file to see the status.
2. Go to the directory `$ORACLE_HOME/ldap/admin` and check the status of replication jobs by running the following command:

   ```
   sqlplus system/password@net_service_name @ldaplogq.sql
   ```

Run this command for each node in the DRG. Issuing this command should result in no rows being selected. If rows are selected containing the status [failed] and error messages, then this means that Oracle9*i* Replication set up failed. In this case, you may:

- Run the script from the beginning
- Consult the troubleshooting chapter in *Oracle9i Replication*
- Determine a solution from error message information by consulting an expert in Oracle9*i* Replication

Task 6: Switch the Sponsor Node to Updatable Mode

To switch the sponsor node to updatable mode:

1. Edit `change_mode.ldif` to the following:

   ```
   dn:
   changetype: modify
   replace: orclservermode
   orclservermode: rw
   ```

2. Run the following commands on the sponsor node:

   ```
   ldapmodify -D "cn=orcladmin" -w welcome -h host_name_of_sponsor_node
   -p  port  -f change_mode.ldif

   oidctl connect=net_service_name server=oidldapd restart
   ```

Note: Task 6 is very similar to Task 3. The only difference is that the `orclservermode` parameter in `change_mode.ldif` is being set back to `rw`, that is, Read-Write, in this step.

Task 7: Start the Directory Replication Server on All Nodes Except the New Node

To start the directory replication server, type the following command:

```
oidctl connect=db_connection_string server=oidrepld instance=1
flags='-h host -p port' start
```

Verify that no directory or replication processes are running on the new node.

Task 8: Load Data into the New Node by Using bulkload

To load data, type the following command:

```
bulkload.sh -connect db_connect_string_of_new_node -generate -load
-restore absolute_path_to_the_ldif_file_generated_by_ldifwrite
```

Task 9: Start LDAP Server on the New Node

To start the LDAP server, type the following command:

```
oidctl connect=db_connect_string_of_new_node server=oidldapd
instance=1 flags='-p port' start
```

Task 10: Configure the LDAP Replication Agreement on the New Node

Run the following command against the new node to add the LDIF file you created in "Task 2: Configure the New Node into the LDAP Replication Group on All the Existing Nodes" on page 15-21:

```
ldapmodify -h host_name_of_the_new_node -p port -f add_node.ldif
```

Task 11: Start the Directory Replication Server on the New Node

To start the directory replication server, type the following command:

```
oidctl connect=db_connect_string_of_new_node server=oidrepld instance=1
flags='-h host_name_of_new_node -p port' start
```

Deleting a Replication Node

At times, you may want to delete a node from a **DRG**. For example, if the addition of a new node did not fully succeed as a result of system errors, then you need to delete that node.

You can delete a replication node from a **DRG** only if there are more than two nodes in the DRG.

To delete a replication node from a directory with fewer than a million entries, follow these steps, each of which is more fully described in this section.

Task 1: Stop the Directory Replication Server on All Nodes

Task 2: Stop All Processes in the Node to be Deleted

Task 3: Delete the Node from the Master Definition Site

Task 4: Start the Directory Replication Server on All Nodes

Task 5: Delete the Node from the Replication Group

Task 6: Restart the Directory Replication Server on the Remaining Nodes

---

Note: Commands shown in the following steps require that the following variables be stored in the corresponding directories:

- Binaries: `$`*`ORACLE_HOME`*`/bin`
- SQL scripts: `$`*`ORACLE_HOME`*`/ldap/admin`
- UNIX scripts: `$`*`ORACLE_HOME`*`/ldap/bin`

Before beginning Task 1, be sure that all three variables are in the path.

Task 1: Stop the Directory Replication Server on All Nodes

To stop the directory replication server, run the following command on each node in the DRG:

```
oidctl connect=net_service_name server=oidrepld instance=1 stop
```

Note: The instance number may vary.

Task 2: Stop All Processes in the Node to be Deleted

Stop the **OID Control Utility** and the **OID Monitor**.

See Also:

- "Stopping an Oracle Directory Server Instance" on page 4-5 for instructions about stopping the OID Control Utility
- "Stopping the OID Monitor" on page 4-3 for instructions about stopping the OID Monitor

Task 3: Delete the Node from the Master Definition Site

From the **MDS**, run the following script:

```
ldaprepl.sh -delnode
```

This script executes these operations:

- It quiesces **ASR** at the MDS and other existing **master sites**.
- It deletes the node from the `orclDirReplGroupDSAs` parameter.
- It verifies that all steps have completed successfully.

As the script runs, it asks for the information in Table 15–2, first for the Master Definition Site then for the node to be deleted.

Table 15–2 Oracle9*i* Replication ***Setup Information***

Information	Description
Host Name of MDS or master site	Name of the computer
Global name	Net service name of the MDS or master site database, as listed in `tnsnames.ora`

Once you have provided that information, the script shows you a table of the information you have provided, and asks for confirmation. If the information is not correct, then press N. The script then starts again at the beginning, asking the same information. If the information is correct and you enter Y, then the script begins configuring the sites.

This process can take a long time, depending on your system resources and the size of your DRG. The script keeps you informed of its progress.

Note: If, for any reason, you must interrupt the process before it is complete, then you must start from the beginning.

Troubleshooting Tip: If the process fails, then do the following:

1. Check the `$`*`ORACLE_HOME/`*`ldap/admin/logs/ldaprepl.log` file to see the status.
2. Go to the directory `$`*`ORACLE_HOME/`*`ldap/admin` and check the status of replication jobs by running the following command:

   ```
   sqlplus system/password@net_service_name @ldaplogq.sql
   ```

Run this command for each node in the DRG. Issuing this command should result in no rows being selected. If rows are selected containing the status [failed] and error messages, then this means that Oracle9*i* Replication set up failed. In this case, you may:

- Run the script from the beginning
- Consult the troubleshooting chapter in *Oracle9i Replication*,
- Determine a solution from error message information by consulting an expert in Oracle9*i* Replication

Task 4: Start the Directory Replication Server on All Nodes

To start the directory replication server, type the following command:

```
oidctl connect=net_service_name server=oidrepld instance=1
flags='-h host -p port' start
```

Task 5: Delete the Node from the Replication Group

Before deleting the node from the replication group, be sure that all of its changes have been applied to the other nodes.

The following example creates an LDIF file, `delete_node.ldif`, and configures it into the replication group on all the existing nodes. Notice that this LDIF file does not include the host name of the node to be deleted.

```
dn: orclagreementid=000001,cn=orclreplagreements
changetype: modify
replace: orcldirreplgroupdsas
orcldirreplgroupdsas: host_name_of_existing_node1
orcldirreplgroupdsas: host_name_of_existing_node2
.
.
.
orcldirreplgroupdsas: host_name_of_existing_node_n
```

Run the following command against each node in the LDAP replication group:

```
ldapmodify -h host_name_of_the_node -p port -f delete_node.ldif
```

Task 6: Restart the Directory Replication Server on the Remaining Nodes

After deleting the node, restart the directory replication server on the remaining nodes for greater efficiency. To do this, type the following command:

```
oidctl connect=db_connection_string server=oidrepld instance=1
flags='-h host -p port' restart
```

Resolving Conflicts Manually

This section contains these topics:

- Monitoring Replication Change Conflicts
- Examples of Conflict Resolution Messages
- Using the Human Intervention Queue Manipulation Tool
- Using the OID Reconciliation Tool

Monitoring Replication Change Conflicts

If a conflict has been written into the log, then it means that the system is not able to resolve it by following its resolution procedure. To avoid further replication change conflicts arising from earlier unapplied changes, it is important to monitor the logs regularly.

To monitor replication change conflicts, examine the contents of the replication log. You can distinguish between messages by their respective timestamps.

Examples of Conflict Resolution Messages

Conflict resolution messages, examples of which are shown below, are logged in the file `oidrepld00.log`. The path for this file is *`ORACLE_HOME/`*`ldap/log`. The result of each attempt to resolve the replication conflict is displayed at the end of each conflict resolution message.

Example 1: An Attempt to Modify a Non-Existent Entry

```
2000/08/03::10:59:05:  ************ Conflict Resolution Message ************
2000/08/03::10:59:05:  Conflict reason: Attempted to modify a non-existent
entry.
2000/08/03::10:59:05:  Change number:1306.
2000/08/03::10:59:05:  Supplier:eastlab-sun.
2000/08/03::10:59:05:  Change type:Modify.
2000/08/03::10:59:05:  Target
DN:cn=ccc,ou=Recruiting,ou=HR,ou=Americas,o=IMC,c=US.
2000/08/03::10:59:05:  Result: Change moved to low priority queue after failing
on 10th retry.
```

Example 2: An Attempt to Add an Existing Entry

```
2000/08/03::10:59:05:  ************ Conflict Resolution Message ************
2000/08/03::10:59:05:  Conflict reason: Attempted to add an existing entry.
2000/08/03::10:59:05:  Change number:1209.
2000/08/03::10:59:05:  Supplier:eastlab-sun.
2000/08/03::10:59:05:  Change type:Add.
2000/08/03::10:59:05:  Target DN:cn=Lou Smith, ou=Recruiting, ou=HR,
ou=Americas, o=IMC, c=US.
2000/08/03::10:59:05:  Result: Deleted duplicated target entry which was created
later than the change entry. Apply the change entry again.
```

Example 3: An Attempt to Delete a Non-Existent Entry

```
2000/08/03::10:59:06:  ************ Conflict Resolution Message ************
2000/08/03::10:59:06:  Conflict reason: Attempted to delete a non-existent
entry.
2000/08/03::10:59:06:  Change number:1365.
2000/08/03::10:59:06:  Supplier:eastlab-sun.
2000/08/03::10:59:06:  Change type:Delete.
2000/08/03::10:59:06:  Target DN:cn=Lou
Smith,ou=recruiting,ou=hr,ou=americas,o=imc,c=us.
2000/08/03::10:59:06:  Result: Change moved to low priority queue after failing
on 10th retry.
```

Using the Human Intervention Queue Manipulation Tool

The human intervention queue manipulation tool enables you to move the changes from the human intervention queue to either the retry queue or the purge queue. Moving the change to the purge queue means that there are no further attempts to re-apply the changelog entry. Perform the following general steps to address changes in the human intervention queue:

1. Shutdown the directory replication server.
2. Analyze the replication log.
3. Use the human intervention queue manipulation tool to move the changes to either the retry queue or the purge queue as described in the following sections.

> **See Also:** "Human Intervention Queue Manipulation Tool Syntax" on page A-41

Using the OID Reconciliation Tool

When the directory replication server encounters inconsistent data, you can use the OID reconciliation tool to synchronize the entries on the consumer with those on the supplier. When you do this, perform the following general steps:

1. Set the supplier and the consumer to read-only mode.
2. Ensure that the supplier and the consumer are in tranquil state. If they are not in a tranquil state, then wait until they have finished updating.
3. Identify the inconsistent entries or subtree on the consumer.
4. Use the OID reconciliation tool to fix the inconsistent entries or subtree on the consumer.
5. Set the participating supplier and consumer back to read-write mode.

> **See Also:** "OID Reconciliation Tool Syntax" on page A-44 for syntax and an explanation of how OID reconciliation tool works.

Identifying a Node as Independent of Its Host

In most deployments, a node in a DRG is uniquely identified by the name of the host where Oracle Internet Directory is installed. However, when there are multiple installations of Oracle Internet Directory on the same host, the host name cannot be a unique node identifier. In this case, you should use the `orclReplicaId` attribute of the Root DSE.

When you identify a node in a DRG by using `orclReplicaId` instead of the host name, follow the steps in this section.

> **Note:** Do not perform any updates on the nodes in the DRG until you have modified the `orclReplicaId` Root DSE attribute on all the nodes.

1. On each node in the DRG, give the `orclReplicaId` a unique value. For example, if there are three nodes on the same computer, and the corresponding directory servers are running on port1, port2 and port3, then you would perform following modifications:

   ```
   ldapmodify -v -h host -p port1  << EOF
   dn:
   changetype: modify
   ```

```
replace: orclreplicaid
orclreplicaid : replica001

ldapmodify -v -h host -p port2  << EOF
dn:
changetype: modify
replace: orclreplicaid
orclreplicaid : replica002

ldapmodify -v -h host -p port3  << EOF
dn:
changetype: modify
replace: orclreplicaid
orclreplicaid : replica003
```

2. After you have modified `orclreplicaid` on all the nodes, perform replication setup as described in "Installing and Configuring Replication" on page 15-2.
3. When you modify the DRG as described in "Modifying Replication Agreement Parameters by Using ldapmodify" on page 15-17, give the `orcldirreplgroupdsas` attribute the same value you assigned to `orclreplicaid`. To use the previous example, you would give the `orcldirreplgroupdsas` attribute the values `replica001`, `replica002`, `replica003`.

> **Note:** Once you have set up replication, do not modify the `orclreplicaId` attribute.

16

Adding a Node to a DRG by Using the Database Copy Procedure

This chapter tells how to add a new node to an existing replicating system by using the database copy procedure, also known as **cold backup**.

> **Note:** Because this procedure involves copying Oracle data files, faster performance depends on the underlying network. If the underlying network is weak, then it may be better to implement the method described in Chapter 15, "Managing Directory Replication", or to physically ship compressed Oracle data files on a medium such as a tape or disk. Consult your local system or network administrator for more details on the network.
>
> Only a person familiar with the Oracle database should implement this procedure.

This appendix contains these topics:

- Assumptions
- Sponsor Directory Site Environment
- New Directory Site Environment
- Tasks To Be Performed on the Sponsor Node
- Tasks To Be Performed on the New Node
- Verification Process

Assumptions

This document assumes that the UNIX directories are created according to Optimal Flexible Architecture (OFA), the set of configuration guidelines for efficient and reliable Oracle databases.

See Also: The Oracle installation guide for your operating system for more information on OFA

Sponsor Directory Site Environment

Set up the environment of the sponsor site. In the example shown throughout this chapter, the host name is rst-sun.

```
Hostname    = rst-sun
ORACLE_BASE = /private/oracle/app/oracle
ORACLE_HOME = /private/oracle/app/oracle/product/8.1.6
ORACLE_SID  = LDAP
LD_LIBRARY_PATH = $ORACLE_HOME/lib
NLS_LANG    = AMERICAN_AMERICA.UTF8
datafile location = /private/oracle/oradata/LDAP
Dump destination =  /private1/oracle/app/oracle/admin/LDAP/pfile,
                    /private1/oracle/app/oracle/admin/LDAP/bdump,
                    /private1/oracle/app/oracle/admin/LDAP/cdump,
                    /private1/oracle/app/oracle/admin/LDAP/udump,
                    /private1/oracle/app/oracle/admin/LDAP/create
```

New Directory Site Environment

Set up the environment for the new directory site. In the example shown throughout this chapter, the new site is on the node named dsm-sun.

```
Hostname = dsm-sun
ORACLE_BASE = /private1/oracle/app/oracle
ORACLE_HOME = /private1/oracle/app/oracle/product/8.1.6
ORACLE_SID  = NLDAP
LD_LIBRARY_PATH = $ORACLE_HOME/lib
NLS_LANG = AMERICAN_AMERICA.UTF8
   datafile location = /private1/oracle/oradata/NLDAP
   Dump destination =  /private1/oracle/app/oracle/admin/NLDAP/pfile,
                       /private1/oracle/app/oracle/admin/NLDAP/bdump,
                       /private1/oracle/app/oracle/admin/NLDAP/cdump,
                       /private1/oracle/app/oracle/admin/NLDAP/udump,
                       /private1/oracle/app/oracle/admin/NLDAP/create
```

Note: After installation of the Oracle database or Oracle directory, you use Oracle Database Configuration Assistant to create data file directories. Create the new directories on the new node under various UNIX partitions as defined by OFA.

Tasks To Be Performed on the Sponsor Node

Complete the following steps on the sponsor node.

1. At the command line prompt execute SQL*Plus.

   ```
   $ sqlplus /nolog
   SQL> connect /as sysdba
   SQL> ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
   ```

   The above command will create a trace file under the user dump destination directory (that is, `/private1/oracle/app/oracle/admin/LDAP/udump`).

   The file will be created in the following format:

   ```
   $ORACLE_SID_ora_processid.trc
   ```

   For example:

   ```
   ldap_ora_4765.trc
   ```

2. Shutdown the LDAP and replication servers and OID Monitor processes. Make sure the ldap and replication servers are stopped before stopping the OID Monitor process.

```
   $ oidctl connect=net_service_name server=oidrepld instance=instance_number
stop
   $ oidctl connect=net_service_name server=oidldapd instance=instance_number
stop
   $ oidmon connect=net_service_name stop
```

   In these commands, *net_service_name* is the net service name in the node's `tnsnames.ora` file.

3. On the remaining nodes, shutdown the LDAP replication server only.

   ```
   $ oidctl connect=net_service_name server=oidrepld instance=instance_number
   stop
   ```

Repeat the above procedure on all nodes except the sponsor node. Specify appropriate net service names for the corresponding nodes.

4. Quiesce **Oracle9i Replication** by running the following script at the **master definition site (MDS)**:

   ```
   ldaprepl.sh -quiesce
   ```

   Enter the Oracle global name for the MDS when prompted.

   > **Note:** This procedure can take place only on the Master Definition Site.

   At this point, other nodes are available for LDAP edits only, but replication will not take place.

5. After quiescing the environment, shutdown the database and Oracle Net Services listener on the sponsor node only:

   ```
   $ lsnrctl [listener_name] stop   (By default listener name is LISTENER)
   $ sqlplus /nolog
   SQL> connect /as sysdba
   SQL> shutdown normal
   SQL> exit
   ```

6. Copy the trace file created under Step 1 to a new file, `newdb.sql`, under the same directory.

   ```
   $ cd $ORACLE_BASE/admin/LDAP/udump
   $ cp ldap_ora_4765.trc newdb.sql
   ```

7. Edit `newdb.sql`, using any text editor, and delete the lines up to START NOMOUNT.

   ```
   CREATE CONTROLFILE REUSE SET DATABASE database_name RESETLOG
   ```

8. Modify the UNIX directory location of the database/logfiles etc. to point to the new node directory. Refer to the sample file `newdb.sql` as follows:

   ```
   Begin newdb.sql
   CREATE CONTROLFILE REUSE SET DATABASE "LDAP" RESETLOGS
   MAXLOGFILES 16
   MAXLOGMEMBERS 2
   MAXDATAFILES 255
   MAXINSTANCES 1
   ```

```
MAXLOGHISTORY 100
LOGFILE
GROUP 1 '/private2/oracle/oradata/NLDAP1/log1_NLDAP.dbf'  SIZE 1M,
GROUP 2 '/private2/oracle/oradata/NLDAP1/log2_NLDAP.dbf'  SIZE 1M
DATAFILE
'/private2/oracle/oradata/NLDAP1/sys0_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/rbs1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/attrs1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/dncat1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/cncat1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/objcl1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/cats1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/default1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/temp1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/iattrs1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/idncat1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/icncat1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/iobjcl1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/icats1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/temp2_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/cats2_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/attrs2_NLDAP.dbf'
;
 End newdb.sql
```

9. Copy the files `initLDAP.ora` and `configLDAP.ora` under `$ORACLE_HOME/dbs` to `initNLDAP.ora` and `configNLDAP.ora` respectively.

```
$cd $ORACLE_HOME/dbs
$cp initLDAP.ora initNLDAP.ora
$cp configLDAP.ora configNLDAP.ora
```

10. Edit the copied file (`initNLDAP.ora`) and comment out the parameter JOB_QUEUE_PROCESS. Change the following parameter:

    `db_name = LDAP`   (If the parameter does not exist in the file initNLDAP.ora, then modify the file configNLDAP.ora)
    `ifile =` *`UNIX_directory_location_of_the_new_config_file/`* `configNLDAP.ora`

11. Edit the copied file `configNLDAP.ora` to change the following parameters:

```
cdump =  UNIX_directory_location_of_the_new_node
udump  = UNIX_directory_location_of_the_new_node
bdump  = UNIX_directory_location_of_the_new_node
control_files = UNIX_directory_location_of_the_new_node
```

12. Edit the `tnsnames.ora` file to include information pertaining to the new node. Refer to the following sample file:

```
Begin tnsnames.ora

ldap1.world =
   (description=
      (address=(protocol=tcp)(host=rst-sun)(port=1521))
      (connect_data=(sid=LDAP))
   )
ldap2.world =
   (description=
      (address=(protocol=tcp)(host=eas-sun10)(port=1521))
      (connect_data=(sid=LDAP))
   )
ldap3.world =
   (description=
      (address=(protocol=tcp)(host=dsm-sun)(port=1521))
      (connect_data=(sid=NLDAP))
   )

End tnsnames.ora
```

13. Copy the file `listener.ora` to `list.bak`. Edit the copied file `list.bak` to include the information pertaining to the new node. Refer to the following sample file:

```
Begin listener.ora

# The KEY value for the IPC protocol may be anything, and
# is not related to either the TCP hostname or database SID.

LISTENER =
  (ADDRESS_LIST =
        (ADDRESS=(PROTOCOL= IPC)(KEY= LDAP))
        (ADDRESS=(PROTOCOL= IPC)(KEY= PNPKEY))
        (ADDRESS=(PROTOCOL= TCP)(Host= dsm-sun)(Port= 1521))
  )
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME= dsm-sun.us.oracle.com)
      (ORACLE_HOME= /private1/oracle/app/oracle/product/8.1.6)
      (SID_NAME = NLDAP)
     )
    (SID_DESC =
      (SID_NAME = extproc)
      (ORACLE_HOME = /private1/oracle/app/oracle/product/8.1.6)
      (PROGRAM = extproc)
    )
  )
STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = OFF

End listener.ora
```

The files `tnsnames.ora` and `listener.ora` can reside under `$ORACLE_HOME/network/admin` or `/var/opt/oracle` or under the directory pointed to by the TNS_ADMIN environment variable.

14. Copy the updated `tnsnames.ora` file to all the nodes. Be careful to copy it to the location of the current `tnsnames.ora` on each node. The file `tnsnames.ora` can be copied to other nodes using FTP. Make sure you transfer the file in ASCII mode.

Prior to copying the file `tnsnames.ora` to the new node, install the Oracle database software on the new node. Also copy the files `list.bak` as `listener.ora` and `sqlnet.ora` from the sponsor node to the new node.

15. Create an archive of all the data files and compress the archived file. For example:

```
$ >oradb.tar
```

This command will create an empty file under a directory. Make sure you have enough space in the partition where the archives will be created.

```
$ find / -name *.dbf -print -exec tar rvf  absolute_path_of_the_directory_
which_contains_oradb.tar {} \;
```

This command will search for all files ending with extension `.dbf` from the root directory. The assumption is that there is only one instance of the database server installed on the node and data files end with `*.dbf` extension.

```
$ find / -name *.log -print -exec tar rvf absolute_path_of_the_directory_
which_contains_oradb.tar
$ compress oradb.tar
```

This procedure is only an example to illustrate the method to back up the files. The Oracle data files will be backed up in the absolute path using this method.

It is a better idea to back up the files from the current directory, so that you have more flexibility when you want to restore the data files. Consult your system administrator before backing up the database.

Tasks To Be Performed on the New Node

Complete the following steps on the new node.

1. Log in to the new node (dsm-sun).
2. Edit the `oratab` file appropriately for the new instance, at all database nodes. See the sample file for syntax.

   ```
   Begin oratab

   NLDAP:/private1/oracle/app/oracle/product/8.1.6:N
   *:/private1/oracle/app/oracle/product/8.1.6:N

   End oratab
   ```

3. Make sure the environment variables are set in the new directory site.
4. Install the Oracle database and Oracle directory server. Perform software only install of the Oracle database and directory server. Installation of Oracle database and directory software can be performed on the new node at any time before the database files are copied to the new machine. Perform post-installation (that is: `root.sh`) activities for the database as well as the Directory server.

   See Also: Oracle9*i* installation documentation

   If you have already performed Oracle database and Directory installation on the new node, then proceed to Step 5.

5. Copy the files `initNLDAP.ora` and `configNLDAP.ora` from the sponsor node (rst-sun) to the new node under the UNIX directory `$ORACLE_BASE/ADMIN/NLDAP/PFILE`. Files can be copied to the new machine using tools such as FTP. Make sure the transfer mode is ASCII.

6. Create a symbolic soft link from `$ORACLE_HOME/DBS TO $ORACLE_BASE/ADMIN/NLDAP/PFILE`.

   ```
    $ ln -s $ORACLE_BASE/admin/NLDAP/pfile/initNLDAP.ora
           $ORACLE_HOME/dbs/initNLDAP.ora
    $ ln -s $ORACLE_BASE/admin/NLDAP/pfile/configNLDAP.ora
           $ORACLE_HOME/dbs/configNLDAP.ora
   ```

7. Copy the archived file created in the sponsor node procedure, using a tool such as FTP. (You created this file in Step 15 on page 16-7.) Set the transfer mode to binary.

   ```
   ftp> open rst-sun
   Connected to rst-sun.us.oracle.com.
   220 rst-sun FTP server (UNIX(r) System V Release 4.0) ready.
   Name (rst-sun:oracle):
   331 Password required for oracle.
   Password:
   230 User oracle logged in.
   ftp> cd /private1/oracle/oradata/LDAP
   250 CWD command successful.
   ftp> binary
   200 Type set to I.
   ftp> mget oradb.tar.Z
   ```

   If the data files are huge (several gigabytes or terabytes) and the network bandwidth is low, then it may be a better idea to physically ship the compressed file on any media, such as tape or disk, from the sponsor to the new node.

8. Copy the file `newdb.sql` created under Step 6 of the sponsor node setup to the background user dump destination directory. You must transfer the file newdb.sql only in ASCII mode. For example:

   ```
    $ cd /private1/oracle/app/oracle/admin/NLDAP/udump
                     (that is::$ORACLE_BASE/admin/SID/udump)
   $ ftp
   ftp> open rst-sun
   ftp> cd /private1/oracle/app/oracle/admin/LDAP/udump
   ftp> mget newdb.sql
   ```

9. At the UNIX shell prompt execute the following commands:

```
$ sqlplus /nolog
SQL> connect /as sysdba
SQL> startup nomount
SQL> @newdb.sql
SQL> shutdown normal
SQL> startup (uncomment the parameter job_queue_process prior to  startup)
SQL>exit
$ lsnrctl start
```

10. Log in to the sponsor node and start up the database and listener on the sponsor node; for example, rst-sun.

```
$ telnet rst-sun
$ sqlplus /nolog
SQL> connect /as sysdba
SQL> startup
SQL> exit
$ lsnrctl start (By default listener name is LISTENER)
$ exit
```

11. If the sponsor node is a master site, then proceed to Step 12.

    If the new node is created by using backup database copy of the MDS, then the master definition catalog needs to be dropped and the underlying Oracle9*i* Replication catalogs must be created. To drop the definition of the MDS from the Oracle9*i* Replication catalog on the new node and add the Oracle9*i* Replication catalogs, execute the following scripts.

```
$ cd $ORACLE_HOME/ldap/admin
$ sqlplus repadmin/repadmin
SQL> @ldapdropmds.sql
SQL> @ldapcreindex.sql
```

    Specify the global name of the new node when prompted.

12. To configure the Oracle9*i* Replication, at the shell prompt, execute the following command:

```
$ ldaprepl.sh -addnode
```

13. Update the LDAP replication agreements to include the new node.

    Sample LDIF file:

    ```
    dn: orclagreementid=000001, cn=orclreplagreements
    changetype: modify
    add: orcldirreplgroupdsas
    orcldirreplgroupdsas: dsm-sun
    ```

14. Start up the LDAP replication server on all the nodes, including new and sponsor nodes.

Verification Process

Log in to the Oracle database by using SQL*Plus and specify the user name as `ODS`, and the password `ods` when prompted.

Check the ods_chg_stat table on all nodes and see if they have correct and identical rows. The ods_chg_stat table should contain (*number of nodes*) x (*number of nodes*) rows. For example, if there were two nodes participating in Oracle9*i* Replication-based replication, and you added a third node, the ods_chg_stat table would contain nine rows, that is, 3 x 3, on each node. The rows are shown in the following table:

Supplier	Consumer	Change Number
Node1	node2	*number 1*
Node1	node3	*number 2*
Node1	node1	*number 3*
Node2	node1	*number 4*
Node2	node2	*number 5*
Node2	node2	*number 6*
Node3	node1	0
Node3	node2	0
Node3	node3	0

The rows with consumer names identical to that of suppliers contain the last changes processed by the outbound change log processing threads at the supplier sides. The rows with different supplier and consumer names contain last change numbers already processed from the suppliers to the consumers in question.

Since Node3 is a new node, there have been no changes supplied by Node3 yet. Therefore, the change numbers for Node3 as supplier are 0.

There may be a time delay before all nodes contain identical rows, but this delay should not be more than two to three minutes.

Part V

Directory Deployment

This part discusses various important deployment considerations. It contains these chapters:

17

Capacity Planning Considerations

Capacity planning is the process of assessing applications' directory access requirements and ensuring that the Oracle Internet Directory has adequate computer resources to service requests at an acceptable rate. This chapter explains what you need to consider when doing capacity planning. It guides you through an example of a directory deployment for an email messaging application in a hypothetical company called Acme Corporation

This chapter contains these topics:

About Capacity Planning

If Oracle Internet Directory and the corresponding Oracle9*i* database are running on the same computer, then these are the configurable resources that capacity planners need to consider:

- I/O subsystem (the type and size)
- Memory
- Network connectivity
- CPUs (speed and quantity)

When you plan to acquire hardware for Oracle Internet Directory, you should ensure that all components—such as CPU, memory, and I/O—are effectively used. Generally, good memory usage and a robust I/O subsystem are sufficient to keep the CPU busy.

Any new installation of the Oracle Internet Directory needs two things to be successful:

- Adequate hardware resources so that the installed system can satisfy user demands at peak load rates
- A well tuned system—hardware and software—that makes the best use of available resources, one that squeezes the maximum performance out of available hardware

We begin by looking at an example of a directory deployment for an email messaging application in a hypothetical company called Acme Corporation. As we examine each component of the capacity plan, we will apply our recommendations to the example of Acme Corporation.

The following terms are used throughout this chapter:

Throughput — The overall rate at which directory operations are being completed by Oracle Internet Directory. This is typically represented as "operations per second."

Latency — The time a client has to wait for a given directory operation to complete

Concurrent clients — The total number of clients that have established a session with Oracle Internet Directory

Concurrent operations — The amount of concurrent operations that are being executed on the directory from all of the concurrent clients. Note that this is not necessarily the same as the concurrent clients because some of the clients may be keeping their sessions idle.

Getting to Know Directory Usage Patterns: A Case Study

The ability to assess the potential load on Oracle Internet Directory is very important for developing an accurate capacity plan. Let us examine the email messaging software employed by our hypothetical company, Acme Corporation. The email messaging software in this example is based on Internet Message Access Protocol (IMAP). There are two main types of software that access Oracle Internet Directory:

- The IMAP clients, which will validate email addresses within the company before sending the mail to the IMAP server. These clients include software programs like Netscape Messenger and Microsoft Outlook.
- The messaging software itself, also called the Mail Transfer Agent (MTA), which will look up the directory to route mail from the outside world to internal mailboxes as well as route internal mails to company-wide distribution lists.

Let us assume that the private aliases and private distribution lists of individual users are also stored in the directory. Let us further make the following assumptions, which will allow us to guess the size of the directory:

Total user population	40,000
Average number of private aliases per person	10
Average number of private distribution lists per person	10
Total number of public distribution lists	4000
Total number of public aliases in the company	1000
Number of attributes in each entry in the directory related to this application	20
Number of cataloged attributes	10

Based on the above assumptions, we can derive the overall count of entries in Oracle Internet Directory as:

User entries	40,000 (these represent the users themselves)
Private aliases of users	40,000 x 10 = 400,000 entries
Private distribution lists of users	40,000 x 10 = 400,000 entries
Company wide distribution lists	4000
Company wide aliases	1000

The above assumptions will yield a directory population of about one million entries. Given the user population and the directory population, let us then analyze usage patterns so that we can derive performance requirements from them. A typical user tends to send an average of 10 emails per day and receives an average of 10 emails a day from the outside world. Assuming that there are, on an average, five recipients for each email being sent by a user, this would result in five directory lookups for each email.

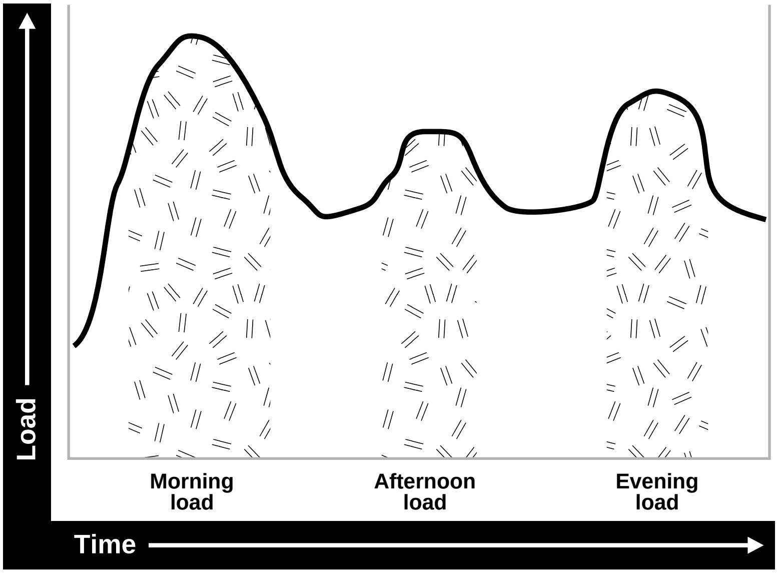
The following table summarizes all the possible directory lookups that can happen in one day:

Type of Directory Lookup	Number of Directory Lookups In One Day
The Mail Transfer Agent (MTA) processing outbound mail from each user	5x10x40,000 = 2,000,000
The MTA processing mails from the outside world	10x40,000 = 400,000
All other directory lookups (like IMAP clients validating certain addresses etc.)	800,000

Summing up, the total number of directory lookups per day would be about 3,200,000 (3.2 million) directory lookups per day. If these directory lookups were spread out uniformly along the day, it would require about 37 directory lookups per second (133,333 lookups per hour). Unfortunately, we will never have this case.

Usage analysis of the current email system over a period of 24 hours shows the pattern illustrated in Figure 17–1.

Figure 17–1 Usage Analysis of Current Email System



The email system and Oracle Internet Directory are maximally stressed in the mornings. There are other usage peaks as well—one close to lunch time, and one near the end of business day. However, it is in the mornings that the Oracle Internet Directory is stressed the most.

Let us assume that 90 percent of all the directory lookups happen during normal working hours. Let us now split up the working hour load into the following categories (assuming an 8 hour workday):

Morning load	65%: 0.90 x 0.65 x 3,200,000 = 1,872,000 lookups for 2 hours (936,000 lookups per hour)
Afternoon load	10%: 0.90 x 0.10 x 3,200,000 = 288,000 lookups for 1 hour (288,000 lookups per hour)
Evening load	20%: 0.90 x 0.20 x 3,200,000 = 576,000 lookups for 2 hours (288,000 lookups per hour)

The above calculations indicate that the Oracle Internet Directory in this case should be designed to handle the peak load of 936,000 lookups per hour.

Now that we know the data-set size as well as the performance requirements, we can now look into individual components of the installation and estimate good values for each.

I/O Subsystem Requirements

This section contains these topics:

- About the I/O Subsystem
- Rough Estimates of Disk Space Requirements
- Detailed Calculations of Disk Space Requirements

About the I/O Subsystem

The I/O subsystem can be compared to a pump that pumps data to the CPUs to enable them to execute workloads. The I/O subsystem is also responsible for data storage. The main components of an I/O subsystem are arrays of disk drives controlled by disk controllers.

It is important to consider performance requirements when you size the I/O subsystem, rather than size based only on storage requirements. Although disk drives have increased in size, the throughput—that is, the rate at which the disk drive pumps data—has not increased in proportion. In sizing calculations for the I/O subsystem, you should use the following factors as input:

- The size of the database
- The number of CPUs on the system
- An initial estimation of the workload on the Oracle Internet Directory
- The rate at which the disk can pump data
- Space needed to stage data prior to load
- Space needed for index creation and sort activities

Given a range of I/O subsystems, you should always opt for the highest throughput drives. Typically, one can maximize the I/O throughput by one or more of the following techniques:

- Striping logical volumes so that the I/O operations use multiple disk spindles
- Putting different tablespaces in different logical and physical disk volumes
- Distributing the disk volumes on multiple I/O controllers

Some guidelines for organizing Oracle Internet Directory-specific data files are provided in Chapter 19, "Tuning Considerations". Depending on the tolerance of disk failures, different levels of Redundant Arrays of Inexpensive Disks (RAID) can also be considered.

Assuming that the decision has been made to get the best possible I/O subsystem, we focus the next section on deriving sizing estimates for the disks themselves.

Rough Estimates of Disk Space Requirements

You can use the following table to derive a rough estimate of the overall disk requirement:

Number of Entries in DIT	Disk Requirements
100,000	450MB to 650MB
200,000	850MB to 1.5GB
500,000	2.5GB to 3.5GB
1,000,000	4.5GB to 6.5GB
1,500,000	6.5GB to 10GB
2,000,000	9GB to 13GB

The data shown in the previous table makes the following assumptions:

- There are about 20 cataloged attributes.
- There are about 25 attributes per entry.
- The average size of an attribute is about 30 bytes.

Going back to our example of Acme Corporation, since our directory population is about one million, this would imply that our disk requirements are approximately 4.5 GB to 6.5 GB. Note that the assumptions made for Acme Corporation regarding the number of cataloged attributes are different, but the previous table should give an approximate figure of the size requirements.

Since the directory may be deployed for a wide variety of applications, these assumptions need not necessarily hold true for all possible situations: There might be cases where the size of attributes is large, the number of attributes per entry is large, extensive use of ACIs has been made, or the number of cataloged attributes is very high. For such cases, we present simple arithmetic procedures in the following section which will allow the planners to get a more detailed perspective of their disk requirements.

Detailed Calculations of Disk Space Requirements

Because Oracle Internet Directory stores all of its data in an Oracle9*i* database, the sizing for disk space is primarily a sizing of the underlying database. Oracle Internet Directory stores its data in the following tablespaces:

OLTS_ATTR_STORE	Stores all of the attributes for all entries in the DIT
OLTS_IND_ATTRSTORE	Stores the indices pertaining to attributes in the directory
OLTS_CT_DN	Stores the distinguished name catalog
OLTS_IND_CT_DN	Stores the indices pertaining to the DN catalog
OLTS_CT_CN	Stores the common name catalog
OLTS_CT_OBJCL	Stores the ObjectClass catalog
OLTS_CT_STORE	Stores all the remaining (including user-defined) catalogs
OLTS_IND_CT_STORE	Stores the indices pertaining to the user-defined catalogs
OLTS_DEFAULT	Stores all of the data pertaining to the administration of the Oracle Internet Directory as well as the data used for replication support
OLTS_TEMP	Used for creating various indices on the tables. It should be large enough so that all index creations can go through.
SYSTEM	Required by Oracle9*i* database for various book-keeping purposes. Typically, its size remains constant at about 300MB.

This section presents simple arithmetic procedures to determine the size requirements of each of the tablespaces shown above. All of the size calculations are based on the following variables:

Variable Name	Description
num_entries	Total number of entries in the directory
attrs_per_entry	Average number of attributes per directory entry
avg_attr_size	Average size of the attribute in bytes
avg_dn_size	Average size of the DN of an attribute in bytes
objectclass_per_entry	Average number of object classes that an entry belongs to
objectclass_size	Average size of the name of each objectclass in bytes
num_cataloged_attrs	Number of cataloged attributes used in the entries
entries_per_catalog	Average number of entries per catalog table. This is required because not all cataloged attributes will be present in all entries in the DIT.
change_log_capacity	Number of changes that we wish to buffer for replication purposes
num_acis	Overall number of ACIs in the directory
num_auditlog_entries	Number of auditlog entries to store in the directory
db_storage_ovhd	Overhead of storing data in tables. This overhead corresponds to the relational constructs as well as operating system specific overhead. A value of 1.3 for this variable would represent a 30 percent overhead. The minimum value for this variable is 1.
db_index_ovhd	Overhead of storing data in indices. This overhead corresponds to the relational constructs as well as the operating system specific overhead. A value of 5 for this variable would represent a 400 percent overhead. The minimum value of this variable is 1.
factor_of_safety	Multiplier for accommodating growth and errors in calculations. A value of 1.3 for this variable would represent a 30 percent factor of safety. The minimum value for this variable is 1.

Using the variables shown in the preceding table, the size of individual tablespaces can be calculated as follows:

Tablespace Name	Size
OLTS_ATTR_STORE	`num_entries * attrs_per_entry * avg_attr_size * db_storage_ovhd`
OLTS_IND_ATTRSTORE	`num_entries * attrs_per_entry * 30`
OLTS_CT_DN	`num_entries * 2 * avg_dn_size`
OLTS_IND_CT_DN	`num_entries * 2 * (avg_dn_size + 30)`
OLTS_CT_CN	`num_entries * avg_dn_size * db_storage_ovhd`
OLTS_CT_OBJCL	`(num_entries * objectclass_per_entry * objectclass_size * db_storage_ovhd) + (num_auditlog_entries * 2 * avg_dn_size * db_storage_ovhd)`
OLTS_CT_STORE	`(entries_per_catalog * num_cataloged_attrs * avg_attr_size * db_storage_ovhd) + (num_entries * objectclass_per_entry * objectclass_size * db_storage_ovhd)`
OLTS_IND_CT_STORE	`(entries_per_catalog * num_cataloged_attrs * avg_attr_size * db_index_ovhd) + (num_entries * objectclass_per_entry * objectclass_size * db_index_ovhd) + (num_acis * 1.5 * avg_dn_size * db_index_ovhd) +       (num_auditlog_entries * 2 * avg_dn_size * db_index_ovhd)`
OLTS_DEFAULT	`(change_log_capacity * 4 * avg_attr_size * db_storage_ovhd * db_index_ovhd) + (num_entries * 5)`
OLTS_TEMP	`(size of OLTS_IND_ATTR_STORE) + (size of OLTS_IND_CT_STORE)`
SYSTEM	`300 MB`

Using the arithmetic operations shown in the preceding table, one can compute the exact space requirements for a wide variety of Oracle Internet Directory deployment scenarios. The sum of the sizes of each of the tablespaces should yield the overall database disk requirement. One can optionally multiply that by the "factor_of_safety" variable to get a figure that can compensate for unforeseen circumstances.

Going back to our example of Acme Corporation, we can assign values to each of the variables based on the requirements stated in previous sections. The following table illustrates the values of each variable introduced in this section for Acme Corporation.

Variable Name	Value
num_entries	1,000,000
attrs_per_entry	20
avg_attr_size	32 bytes
avg_dn_size	40 bytes
objectclass_per_entry	5 (each entry belongs to an average of 5 object classes)
objectclass_size	10 bytes
num_cataloged_attrs	10
entries_per_catalog	1,000,000
change_log_capacity	80,000 changes (2 per user)
num_acis	80,000 ACIs (2 per user)
num_auditlog_entries	1000
db_storage_ovhd	1.4 (40% overhead)
db_index_ovhd	5.0 (400% overhead)
factor_of_safety	1.5 (50% factor of safety)

If we now plug these values into the equations described earlier, we get the following values:

Tablespaces Name	Size in Bytes	Size in MB	Size in MB (with factor of safety)
OLTS_ATTRSTORE	896000000	875	1313
OLTS_IND_ATTRSTORE	600000000	586	879
OLTS_CT_DN	80000000	78	117
OLTS_IND_CT_DN	140000000	137	205
OLTS_CT_CN	56000000	55	82
OLTS_CT_OBJCL	70112000	68	103
OLTS_CT_STORE	518000000	506	759
OLTS_IND_CT_STORE	1874400000	1830	2746
OLTS_DEFAULT	76680000	75	112
OLTS_TEMP	2474400000	2416	3625
SYSTEM	307200000	300	450
Total Size	**7092792000**	**6927**	**10390**

The table above shows that the estimated size of the database for Acme Corporation would be about 6.9 GB. With a 50 percent factor of safety, this would jump to 10.4GB. If all of the data is being loaded in bulk, then the bulkload tool of Oracle Internet Directory would require an additional 50 percent of space occupied by the database to store its temporary files. For Acme Corporation, this would add about 2.25 GB to 3.35 GB to the total space requirement.

Memory Requirements

Memory is used for a number of distinct tasks by any database application, including Oracle Internet Directory. If memory resources are insufficient for any of these tasks, the bottleneck causes the CPUs to work at lower efficiency and system performance to drop. Furthermore, memory usage increases in proportion to the number of concurrent connections to the database and the number of concurrent users of the directory.

The memory available to processes comes from the virtual memory on the system, which is somewhat more than available physical memory. If the sum of all active memory usage exceeds the available physical memory on the system, the operating system may need to store some of the memory pages on disk. This is called paging. Paging can degrade performance if memory is too oversubscribed. Generally, you should not exceed 20 percent over-subscription of physical memory. If paging occurs, you need either to scale back memory usage by processes or to add more physical memory. Keep in mind the trade-offs: There are physical limits to the amount of memory you can add, but scaling back on per-process memory usage can significantly degrade performance.

The main consumer of memory is the database buffer cache within the **System Global Area (SGA)**. The more memory allocated to this, the better will be the buffer cache hit ratio. A good buffer cache hit ratio will result in good database performance which in turn will result in good performance of the Oracle Internet Directory.

> **See Also:** Chapter 19, "Tuning Considerations" for further information on SGA tuning

The following table gives minimum memory requirements for different directory configurations:

Directory Type	Entry Count	Minimum Memory
Small	Less than 600,000	512 MB
Medium	600,000 to 2,000,000	1 GB
Large	Greater than 2,000,000	2 GB

Going back to our example of Acme Corporation, the number of entries in the directory are close to 1,000,000 (1 million). Oracle Corporation recommends choosing the 2 GB option in order to maximize performance.

Network Requirements

The network is rarely a bottleneck in most installations. However serious consideration must be given to it during the capacity planning stage. If the clients do not get adequate network bandwidth to send and receive messages from Oracle Internet Directory, the overall throughput will seem to be very low. For example, if we have configured Oracle Internet Directory to service 800 search operations per second, but the computer running the Oracle directory server is only accessible through a 10 Mbps network (10-Base-T switched ethernet), and we have only 60 percent of the bandwidth available, then the clients will only see a throughput of 600 search operations a second (assuming each search operation causes 1024 bytes to be transferred on the network). The following table shows the maximum possible throughput (in operations per second) for two types of operations (one requiring a transfer of 1024 bytes the other requiring a transfer of 2048 bytes) for two types of networks, 10 Mbps & 100 Mbps, at different rates of bandwidth availability:

Percent Available Bandwidth	Operations/sec 1024 bytes		Operations/sec 2048 bytes	
	10 Mbps	100 Mbps	10 Mbps	100 Mbps
30	300	3000	150	1500
40	400	4000	200	2000
50	500	5000	250	2500
60	600	6000	300	3000
70	700	7000	350	3500
80	800	8000	400	4000
90	900	9000	450	4500

In some cases, it may also be important to consider the network latency of sending a message from a client to the Oracle directory server. In some WAN implementations, the network latencies may become as high as 500 milliseconds, which may cause the clients to time out for certain operations. In summary, given a range of networking options, the preferred choice should always be for highest bandwidth, lowest latency network.

Going back to the example of Acme Corporation, their peak usage rate is 936,000 lookups per hour which results in an equivalent number of lookup operations to the directory. This requires about 260 directory operations per second. Assuming that each operation results in a transfer of 2 KB of data on the network, this would imply that we should have a 100 Mbps network or at least 60 percent bandwidth available on a 10 Mbps network. Since the 100 Mbps network will typically have a lower latency, we will chose that over the 10 Mbps network.

CPU Requirements

This section contains these topics:

- CPU Configuration
- Rough Estimates of CPU Requirements
- Detailed Calculations of CPU Requirements

CPU Configuration

The CPU sizing for Oracle Internet Directory is directly a function of the user workload. The following factors will determine CPU configuration:

- The number of concurrent operations you want to support. This will be directly dependent on the number of users performing operations simultaneously.
- The acceptable latency of each operation. For example, in an email application, a latency per operation of 100 milliseconds might be desirable, but in most cases a latency of 500 milliseconds might still be acceptable.

CPU resources can be added to a system as the workload increases, but these additions seldom bring linear scalability to all operations since a lot of operations are not purely CPU bound. We classify the processing power of a computer by a performance characteristic that is commonly available from all vendors, namely, SPECint_rate95 baseline. This number is derived from a set of integer tests and is available from all system vendors as well as the SPEC Web site (http://`www.spec.org`).

Note: SPECint_rate95 should not be confused with the regular SPECint95 performance number. The SPECint95 performance number gives an idea of the integer processing power of a particular CPU (for systems with multiple CPUs, this number is typically normalized). The SPECint_rate95 gives the integer processing power of an entire system without any normalization.

Because Oracle Internet Directory makes efficient use of multiple CPUs on an SMP computer, we chose to categorize computers based on their SPECint_rate95 numbers. Even within SPECint_rate95 we chose the baseline number as opposed to the commonly advertised result. This is because the commonly advertised result is actually the peak performance of a computer, whereas the baseline number represents the performance in normal circumstances.

Rough Estimates of CPU Requirements

Since Oracle Internet Directory is typically co-resident with the Oracle9*i* database, we recommend at least a two-CPU system. We give the following rough estimates based on the level of usage of Oracle Internet Directory:

Usage	Num CPUs	SPECint_rate95 baseline	System
Departmental	2	60 to 200	Compaq AlphaServer 8400 5/300 (300Mhz x 2)
Organization wide	4	200 to 350	IBM RS/6000 J50 (200MHz x 4)
Enterprise wide	4+	350+	Sun Ultra 450 (296 MHz x 4)

Detailed Calculations of CPU Requirements

It is difficult to determine the CPU requirements for all operations at a given deployment site since the amount of CPU consumed depends upon several factors, such as:

- The type operation: base search, subtree search, modify, add etc.
- If SSL mode is enabled or not. SSL consumes an additional 15 to 20 percent of CPU resources.
- The number of entries returned for a search
- The number of access control policies that need to be checked as part of a search

In most of the cases, except SSL, we can expect that there is a large latency between the Oracle Internet Directory server process and the database. When a thread in the Oracle Internet Directory server process is waiting for the database to respond, other threads within the Oracle Internet Directory server process can be put to work by other client requests needing LDAP server specific processing. As a result, for any mix of operations, one can always come up with a combination of concurrent clients and Oracle Internet Directory server processes that will result in 100 percent CPU utilization. In this case, the CPU becomes the bottleneck.

Given this fact, we have taken the operation that consumes the smallest number of CPU cycles: a base search and estimated the number of concurrent operations at which we peaked on CPU usage on various computers. We then correlated this to SPECint_rate95 baseline number of the computers. With this correlation, given a certain amount of concurrency on the user load, one can find a lower bound on the processing power required by Oracle Internet Directory. The following formula gives the concurrency to SPECint_rate95 baseline number for this release of Oracle Internet Directory:

```
SPECint_rate95 baseline = 6 * (concurrent base search operations)
```

For example, if we need a computer that is capable of handling 50 concurrent base search operations before saturating the CPU, we would require a computer that has a SPECint_rate95 baseline rating of about 300.

Taking this number as the baseline, we can find the CPU requirements of other operations if we express them as some factor of the base search operations. The following factors may be used in addition to others:

- If using SSL mode, multiply CPU requirements by a factor of 1.2.
- If one is fetching a lot of entries in each search, multiply CPU requirements by a factor of (1 + 0.2* *num_entries_per_search*).
- Incorporate a factor of safety of 20 percent to 30 percent (multiply by 1.2 to 1.3).

Going back to our example of Acme Corporation, let us assume that we want adequate CPU resources to support about 100 concurrent operations. Assuming that each search returns 1.5 entries, and adding a factor of safety of 20 percent, our preliminary estimate of the CPU requirements would be:

```
SPECint_rate95 baseline = 6 * 100 * (1 + 0.2 * 1.5) * 1.2
= 600 * 1.3 * 1.2
= 936
```

Looking at the available systems from the SPEC Web site (`http://www.spec.org`) we can see that the following computer configurations would be the smallest configurations that should be considered.

The next table shows some of the computers that Acme Corporation can consider using for Oracle Internet Directory.

Company	Model	CPUs	CPU type	SPECint95_rate baseline
Sun Microsystems	ES 4002	12	250MHz UltraSPARC II	943
Siemens Nixdorf	RM600 Model E60	8	250 MHz R10000	970
Hewlett-Packard	HP SPP1600	32	120 MHz PA-RISC 7200	996
SGI	Origin2000	8	250 MHz MIPS R10000	1001
Data General Corporation	AViiON AV 20000	16	Pentium Pro (200 MHz)	1007
Sun Microsystems	Sun Enterprise 3500	8	400MHz UltraSPARC II	1011
Sun Microsystems	Sun Enterprise 3500	8	400MHz UltraSPARC II	1030
Hewlett-Packard	HP 9000 Model N4000	4	440 MHz PA-RISC 8500	1093
Hewlett-Packard	HP 9000 Model T600	12	180MHz PA-RISC 8000	1099
Siemens AG	RM600 Model E80	8	285 MHz R12000	1103
Compaq Corporation	AlphaServer 8400 5/440	12	437 MHz 21164	1146
Compaq Corporation	AlphaServer 8400 5/625	8	612 MHz 21164	1153
SGI	origin2000	16	195 MHz MIPS R10000	1182
Sun Microsystems	Sun Enterprise 4000	12	336MHz UltraSPARC II	1211

Summary of Capacity Plan for Acme Corporation

In the preceding sections, we have described various components involved in capacity planning and have also shown how each of them would apply to an Oracle Internet Directory deployment at a hypothetical company named Acme Corporation. In this section we give a quick summary of all of the recommendations made. Following were the initial assumptions:

- Overall directory size: 3,200,000 entries (3.2 million)
- Number of users: 40,000
- Type of application: IMAP messaging
- Peak search rate: 260 searches/sec
- Concurrent usage rate for best CPU utilization: 100

Based on the above requirements and further assumptions, we developed the following recommendations:

- Disk space: 7 GB to 11 GB
- Memory: 2 GB
- Network: 100 Base-T
- CPU: something that has a SPECint_rate95 of at least 936.

Several simplifying assumptions were made so that the sizing calculations could be more intuitive.

18

High Availability And Failover Considerations

This chapter discusses the high availability and failover features and deployment guidelines for Oracle Internet Directory. It contains these topics:

- About High Availability and Failover for Oracle Internet Directory
- Oracle Internet Directory and Oracle9i Technology Stack
- Failover Options on Clients
- Failover Options in the Public Network Infrastructure
- Availability and Failover Capabilities in Oracle Internet Directory
- Failover Options in the Private Network Infrastructure
- High Availability Deployment Examples

> **See Also:** Part VI, "The Directory and Clusters" for information about high availability and failover in clustered environments

About High Availability and Failover for Oracle Internet Directory

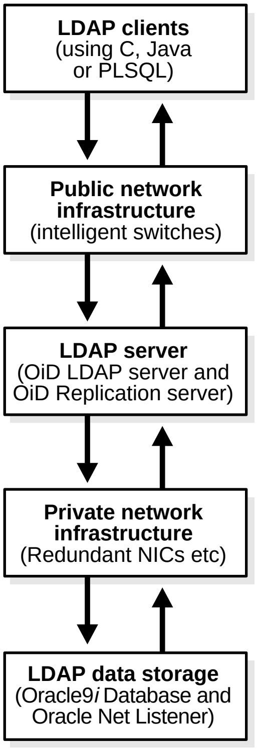
Oracle Internet Directory is designed to address the deployment needs of mission critical applications requiring a high degree of system availability. To achieve a high degree of availability, all components in the system must facilitate redundancy, and all interfaces must facilitate failure recognition and recovery, called **failover**. In addition, integration of application independent network failover capabilities in the overall deployment is also essential to achieve overall system availability.

Oracle products are commonly targeted for high availability environments and hence necessary capabilities are built into all layers of the Oracle technology stack described on page 18-2. Typically, it is not necessary to employ every failover capability in every component. This chapter describes the availability and failover features of various components in the Oracle Internet Directory technology stack, and provides guidelines for exploiting them optimally for typical directory deployment.

Oracle Internet Directory and Oracle9*i* Technology Stack

Figure 18–1 gives an overview of the various components of the Oracle Internet Directory stack. Stack communication between separate computers occurs by passing information from one node to the other through several layers of code. Information descends through layers on the client side. It is then packaged for transport across a network medium. The information then proceeds up the stack on the server side where it is translated and understood by the corresponding layers.

Figure 18–1 Oracle Internet Directory/Oracle9i Technology Stack



You can build sufficient fault tolerance mechanisms into each of the layers to ensure maximum availability of the product. In the following sections we describe some of the high availability options available to our customers in each of the layers shown above.

Failover Options on Clients

Incorporating enough intelligence in the clients so that they can failover to alternate Oracle directory servers in case the primary Oracle directory server fails is a good option in some cases. This requires the clients to cache alternate server information and use it upon recognizing connectivity loss. This method of guaranteeing availability is viable only for deployments in which one has full control over the type of clients accessing the directory.

This section contains these topics:

- Alternate Server List from User Input
- Alternate Server List from the Oracle Internet Directory Server

Alternate Server List from User Input

The clients can be designed to take input from the user on the list of alternate Oracle directory servers so that the clients can automatically failover in the event of a failure of the primary server. However, as the number of clients increases, this option would not scale very well in terms of administration of client installations.

Alternate Server List from the Oracle Internet Directory Server

Oracle Internet Directory supports a DSE root attribute called `AltServer`. This is an LDAP Version 3 standard attribute and is to be maintained by the directory administrator. It is expected to have references to other Oracle directory servers in the system with the same set of naming contexts as that of the local server. When connectivity to the local server is lost, clients have the option of accessing one of the servers listed in this attribute. This option requires explicit administrative action to maintain this attribute.

See Also:

- "Managing Attributes by Using Oracle Directory Manager" on page 7-17 and "Managing Attributes by Using Command-Line Tools" on page 7-29 to set the `AltServer` attribute

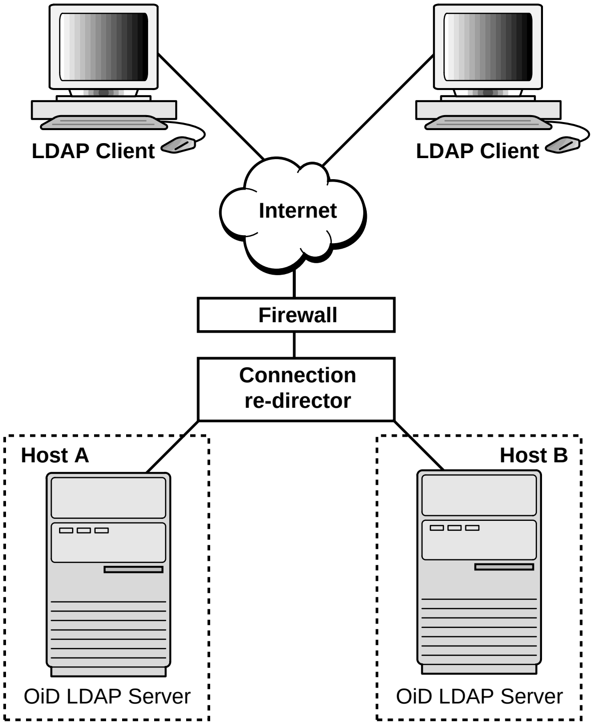
Failover Options in the Public Network Infrastructure

The network used to access Oracle Internet Directory services is called the Public Network Infrastructure. Providing network level load balancing and failover measures (connection re-direction) in the Public Network Infrastructure are highly recommended since these measures provide a high degree of flexibility and transparency to the application clients.

If the Oracle Internet Directory services are accessed from the Internet, this would include a couple of high speed links (T1 to T3) and an intelligent TCP/IP level connection re-director. If the Oracle Internet Directory services are accessed from an Intranet, this would include high speed LAN connections to the server computers running the Oracle directory server and an intelligent TCP/IP level connection re-director. In both cases, there would be more than one computer serving LDAP requests so that failure of one Oracle directory server computer would not affect availability.

Figure 18–2 illustrates a typical Internet deployment of Oracle Internet Directory with network-level failover enabled.

Figure 18–2 Network-Level Failover



In Figure 18–2, the Oracle directory servers (OiD LDAP Servers) can be connected to either the same back-end database or different back-end databases. In this deployment, network-level connection redirection can be accomplished by both hardware and software solutions.

This section contains these topics:

- Hardware-Based Connection Redirection
- Software-Based Connection Redirection

Hardware-Based Connection Redirection

Hardware-based connection redirection technology is available from several vendors. These redirection devices connect directly to the Internet and can route requests among several server computers. They can also detect computer failures and stop routing requests to the failed computer. This feature guarantees that new connections from clients will not be routed to a failed computer. When a computer comes back, the device detects it and starts routing new requests to it. These devices also perform some load balancing, which makes sure that client requests are uniformly distributed.

Some of the vendors providing hardware based re-direction technologies are:

- Accelar Server Switches from Nortel Networks
- Local Director from Cisco
- BIG/ip from F5 Labs Inc.
- Hydra from HydraWEB Technologies
- Equalizer from Coyote Point Systems

Software-Based Connection Redirection

The software-based solutions essentially work in the same manner as their hardware counterparts. Some of the currently available solutions include Dispatch from Resonate and Network Dispatcher from IBM.

Availability and Failover Capabilities in Oracle Internet Directory

Multimaster replication makes it possible for the directory system to be available for both access and updates at all times, as long as at least one of the nodes in the system is available. When a node comes back online after a period of unavailability, replication from the existing nodes will resume automatically and cause its contents to be synchronized transparently.

Any directory system with high availability requirements should always employ a network of replicated nodes in multimaster configuration. A replica node is recommended for each region that is separated from others by a relatively low speed or low bandwidth network segment. Such a configuration, while allowing speedy directory access to the clients in the same region, also serves as a failover arrangement during regional failures elsewhere.

Failover Options in the Private Network Infrastructure

The Private Network Infrastructure is the network used by Oracle Internet Directory and its back-end components to communicate with each other. In cases where Oracle Internet Directory is deployed on the Internet, Oracle Corporation recommends that this network be physically different from the network used to serve client requests. In cases where Oracle Internet Directory is deployed over an Intranet, the same LAN may be used, but Oracle Internet Directory components should have dedicated bandwidth with the help of a network switch. Because Oracle Internet Directory depends on the Private Network Infrastructure for its communications, you must take adequate precautions to guarantee availability in the event of failures in the Private Network. Some of the options available in this area are:

- IP Address Takeover (IPAT)
- Redundant Links

IP Address Takeover (IPAT)

IP address takeover feature is available on many commercial clusters. This feature protects an installation against failures of the Network Interface Cards (NICs). In order to make this mechanism work, installations must have two NICs for each IP address assigned to a server. Both the NICs must be connected to the same physical network. One NIC is always active while the other is in a standby mode. The moment the system detects a problem with the main adapter, it immediately fails over to the standby NIC. Ongoing TCP/IP connections are not disturbed and as a result clients do not notice any downtime on the server.

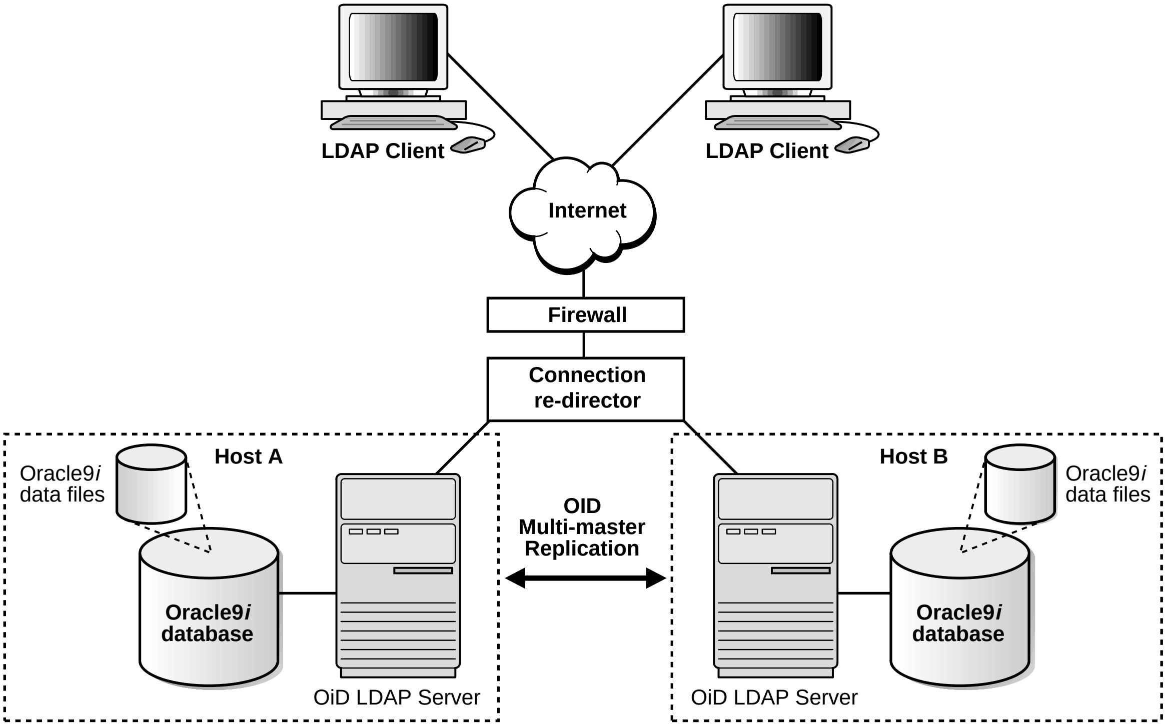
Redundant Links

Since all networks (with the exception of wireless networks) are comprised of wires going from one location to the other, there is a distinct possibility that someone might unintentionally disconnect a wire that is used to link a client computer to a server computer. If you want to take such precautions, use NICs and hubs/switches that come with the capability to use redundant links in case of a link level failure.

High Availability Deployment Examples

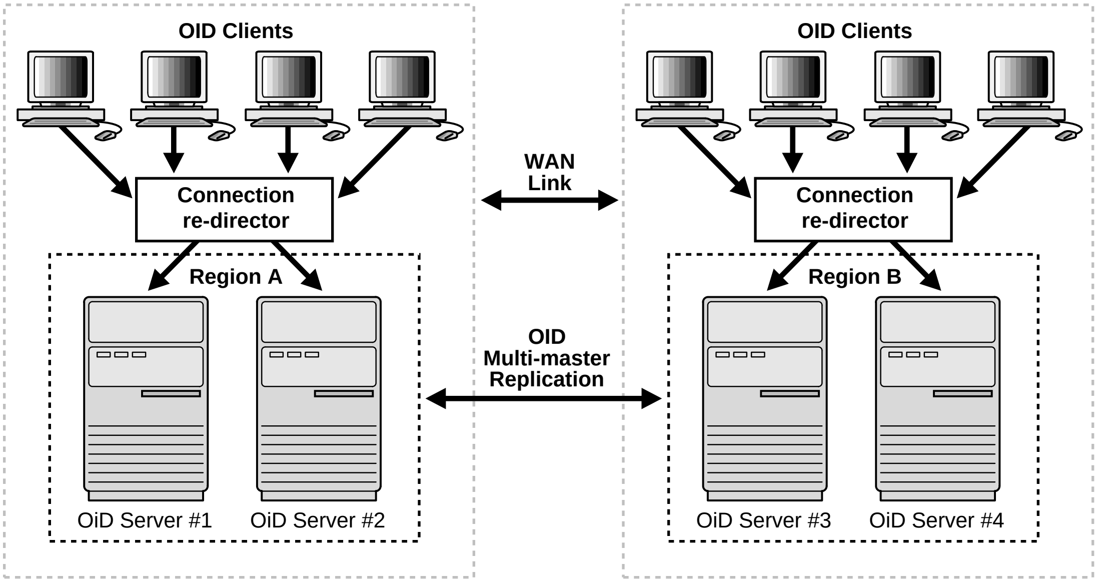
In Figure 18–3, the database and Oracle directory server (OiD LDAP Server) are co-resident on the same computer. Changes made on one directory server instance are reflected on the second directory server instance through multimaster replication. When a failure of the directory server or database server on a particular node occurs, it is elevated to a computer failure so that the connection redirector will stop handing off connections to the computer on which there was a failure.

Figure 18–3 Deployment Example (Two Oracle Internet Directory Nodes in Replication)



As Figure 18–4 illustrates, each of the regions can be set up with two Oracle Internet Directory nodes replicating between each other. This configuration is typical of global directory networks deployed by large enterprises where each of the regions above could potentially represent a continent or a country.

Figure 18–4 Deployment Example 2



19

Tuning Considerations

Once you have completed capacity planning as described in Chapter 17, "Capacity Planning Considerations", and you have acquired the necessary hardware, then you must ensure that the combined hardware and software are yielding the desired levels of performance. This chapter gives guidelines for tuning an Oracle Internet Directory installation. It contains these topics:

- About Tuning
- Tools for Performance Tuning
- CPU Usage Tuning
- Memory Tuning
- Disk Tuning
- Database Tuning
- Performance Troubleshooting

About Tuning

The two main performance metrics for any installation of Oracle Internet Directory are:

- The average latency of individual operations at peak load

  This is the time for each operation to complete.

- The overall throughput of Oracle Internet Directory expressed in operations per second at peak load

  This is the rate at which an instance of Oracle Internet Directory is capable of completing client operations

If the performance tests yield poor results, the performance problems may be identified and fixed using the information provided in the following sections.

Tools for Performance Tuning

Knowledge of the following tools is recommended for Solaris and most other UNIX operating systems:

Tool	Description
top	Displays the top CPU consumers on a system
vmstat	Shows running statistics on various parts of the system including the Virtual Memory Manager
mpstat	Shows an output similar to vmstat but split across various CPUs in the system. This is available on Solaris only.
iostat	Shows the disk I/O statistics from various disk controllers

Knowledge of the following tools is recommended for Windows NT:

Tool	Description
Windows NT Performance Monitor	Provides a customized view of the events in the system
Windows NT Task Manager	Provides a high level output (like ‘top’ on UNIX) of the major things happening in the system.

Knowledge of the following tools is recommended for Oracle9*i*:

- `utlbstat.sql` and `utlestat.sql`
- The ANALYZE function in the DBMS_STATS package

> **See Also:**
>
> - *Oracle9i Database Reference* for information about `utlbstat.sql` and `utlestat.sql`
> - *Oracle9i Database Concepts* for information about the ANALYZE function in the DBMS_STATS package

In addition to the operating system tools, the LDAP applications being used in a customer environment must be able to provide latency and throughput measurement.

In addition, the Database Statistics Collection Tool (oidstats.sh), located at `$ORACLE_HOME/ldap/admin`, is provided to analyze the various database 'ods' schema objects to estimate the statistics.

> **See Also:** "OID Database Statistics Collection Tool Syntax" on page A-47

CPU Usage Tuning

The CPU is perhaps the most important resource available for any software. While Chapter 17 gives a rough estimate of the required CPU horsepower for a given application load, sometimes insufficient tuning can cause inefficient use of the CPU resources. Consider tuning CPU resources if either of the following cases is true:

- At peak loads the CPU is 100 percent utilized.
- At peak loads the CPU is underutilized, there is a significant amount of idle time in the system, and this idle time cannot be eliminated at even higher loads.

Internal benchmarks show that Oracle Internet Directory performs best when approximately 70 to 75 percent of the CPU resources are consumed by Oracle Internet Directory processes, and the remaining (about 25 to 30 percent) are consumed by the Oracle foreground processes corresponding to the database connections. While monitoring CPU usage, it is also important to monitor the percentage of time spent in the system space compared to user space. Internal benchmarks show best throughput numbers at about 85 percent user and 15 percent system time.

This section contains these topics:

- Tuning CPU for Oracle Internet Directory Processes
- Tuning CPU for Oracle Foreground Processes
- Taking Advantage of Processor Affinity on SMP Systems
- Other Alternatives for a CPU Constrained System

Tuning CPU for Oracle Internet Directory Processes

The demands placed by Oracle Internet Directory processes on the CPU can be controlled by the ORCLSERVERPROCS and ORCLMAXCC parameters. This table lists suggested values for these parameters for various client loads:

Parameters	500 Concurrent LDAP Clients	1000 Concurrent LDAP Clients	1500 Concurrent LDAP Clients	2000 Concurrent LDAP Clients
Server processes ORCLSERVERPROCS	10 to 15	20 to 30	30 to 40	40 to 60
Database connections ORCLMAXCC	10 to 15	15 to 20	15 to 20	15 to 20

If we take the example of 500 concurrent clients, a value of 10 for ORCLSERVERPROCS with a value of 15 for ORCLMAXCC will result in the following configuration:

- There will be ten server processes created.
- Each server process will spawn fifteen worker threads that will do the actual work.
- Each server process will also maintain a pool of sixteen database connections (15+1) that will be shared among the worker threads.

Tuning Oracle Internet Directory Processes When CPU Is 100 Percent Utilized

If the CPU usage of the system is at 100 percent, further tuning of the Oracle Internet Directory processes should be considered if both of the following conditions are met:

- At peak loads, Oracle Internet Directory processes consume more than 70 percent of all available CPU resources.
- At peak loads, the overall percentage of time spent in the 'system' or 'kernel' space is greater than 20 percent, and the percentage of time spent in the 'user' time is less than 80 percent.

This condition indicates that the system has too manyOracle Internet Directory server processes and database connections configured. This results in several processes or threads contending for the same CPU resources. As a result, the computer wastes a great deal of time context-switching among runnable tasks. To avoid this, one must systematically decrease the values of ORCLSERVERPROCS and ORCLMAXCC until the best performance for the peak load is achieved and the system and user time are split up as follows:

- User time: 85 percent or higher
- System time: 15 percent or lower

Tuning Oracle Internet Directory Processes When CPU Is Under-Utilized

If the CPU usage at peak loads is not at 100 percent and the system is idle for a large percentage of the time (that is, more than 5 percent), this indicates that Oracle Internet Directory processes are under-configured and are not making the best utilization of the CPU resources. To solve this problem, one must systematically increase the values of ORCLSERVERPROCS and ORCLMAXCC until the CPU utilization reaches 100 percent and the system and user time are split up as follows:

- User time: 85 percent or higher
- System time: 15 percent or lower

Tuning CPU for Oracle Foreground Processes

Tuning of CPU resources for Oracle Foreground processes should be considered only if both of the following conditions are met:

- The CPU usage is close to 100 percent at peak loads.
- Oracle foreground processes consume more than 30 percent of all available CPU resources.

If Oracle foreground processes are consuming excessive CPU, it implies that the queries that Oracle Internet Directory is making against the database are using too many CPU cycles. Although there is very little control available to the users on the types of underlying operations performed by the database, the following should be attempted:

- Database statistics on all of the tables and indices associated with the ODS user on the database must be collected using the ANALYZE command. This helps the cost based optimizer make better execution plans for the queries generated by Oracle Internet Directory.
- If the ANALYZE fails to produce better results, and the LDAP queries used have a lot of filters in them, then a simple reorganization of the order in which the filters are specified (with the most specific filter in the beginning and the most generic filter at the end) helps reduce the CPU consumption of the Oracle foreground processes.

Taking Advantage of Processor Affinity on SMP Systems

Several Symmetric Multi-Processor (SMP) systems offer the capability to bind a particular process to a particular CPU. While it is generally a good idea not to bind any process to any processor, it may improve performance if the following conditions are met:

- The CPU utilization of the entire system is close to 100 percent.
- There are more than two CPUs on the computer.
- Oracle Internet Directory processes consume around 70 to 75 percent of the CPU resources.
- The database processes consume around 25 to 30 percent of the CPU resources.

Under the conditions noted above, allowing the database foreground process to run on any CPU can potentially cause many hardware cache misses for other tasks. This is because the database processes need to reference a large amount of data as part of their regular execution, and this often exceeds the limits of L2 caches available on most systems. As a result, when the database process executes on a CPU, most of L2 cache contains pages from the **System Global Area (SGA)**. If a task switch occurs and an Oracle Internet Directory process is activated, all of its fetches from memory will be much slower because the task preceding it on the processor dirtied the L2 cache.

Restricting all of the Oracle foreground processes to execute on only one processor avoids many of the cache misses for Oracle Internet Directory processes. This, in turn, improves the overall performance.

Other Alternatives for a CPU Constrained System

If none of the tips stated in the preceding sections solve CPU related performance problems, the following options are available:

- Upgrade the processing power of the computer, that is, add more CPUs or replace slower CPUs with faster ones.
- Keep the Oracle directory server and the associated Oracle9*i* database on separate computers.

Memory Tuning

After the CPU, memory is the next most important thing to tune. The primary consumer of memory in an Oracle Internet Directory installation is the Oracle9*i* database. Make the SGA of the back-end database large enough while leaving room for Oracle Internet Directory and Oracle processes to operate their private stacks and heaps. This section provides some details on determining various components of the SGA.

This section contains these topics:

- Tuning the System Global Area (SGA) for Oracle9i
- Other Alternatives for a Memory-Constrained System

Tuning the System Global Area (SGA) for Oracle9*i*

The SGA should be sized based on the available physical memory on the system running Oracle9*i*.

> **See Also:** *Oracle9i Database Performance Guide and Reference* for more information on determining appropriate sizes for the SGA. This book tells how to ensure that the SGA size does not cause increased paging swapping activity. The latter is very detrimental to performance.

Once the available size of the SGA is determined, two primary tuning items need to be considered:

- Size of the shared pool
- Size of the buffer cache

An initial estimate for the shared pool size is .5 MB per concurrent database connection determined above.

If this estimate consumes more than 30 percent of the total SGA, use 30 percent of the total SGA instead.

Divide 60 percent of the remaining available SGA size by the block size for the database and use this value for the number of DB_BLOCK_BUFFERS. Both of these values should be initial estimates and can be refined using BSTAT/ESTAT and other RDBMS monitoring tools to determine more accurate sizes for best performance.

Other Alternatives for a Memory-Constrained System

If there is insufficient memory to run both the database and the Oracle directory server on the same computer, then one can put the database on a different computer.

Disk Tuning

Balancing Disk I/O is an important consideration in overall RDBMS, and hence Oracle Internet Directory performance. Typically, one can maximize the I/O throughput by using one or more of the following techniques:

- Striping logical volumes so that the I/O operations use multiple disk spindles
- Putting different tablespaces in different logical and physical disk volumes
- Distributing the disk volumes on multiple I/O controllers

> **See Also:** *Oracle9i Database Performance Guide and Reference* for general information about balancing and tuning disk I/O

This section contains these topics:

- Balancing Tablespaces
- RAID

Balancing Tablespaces

The Oracle Internet Directory schema is distributed among several tablespaces at installation time for ease of maintenance and performance. Each tablespace contains a grouping of Oracle Internet Directory schema objects appropriate for co-location on disk storage. As available, it is also beneficial to distribute the following objects onto separate logical disks.

> **See Also:** "RAID" on page 19-10 for more discussion about logical disks

Separate the following:

- OLTS_ATTRSTORE and OLTS_IND_ATTRSTORE

  Separating the attribute store table from its index
- OLTS_CT_DN and OLTS_IND_CT_DN

  Separating the DN catalog from its index
- OLTS_xxxx and OLTS_IND_xxxx

  (Empirically, separate the storage tablespace from the associated index)

- OLTS_IND_ATTRSTORE and OLTS_IND_CT_DN

  Alternating the attribute store and DN catalog indexes. This helps even if there are only two logical disks available (one containing OLTS_CT_DN and OLTS_IND_ATTRSTORE and the other containing OLTS_IND_CT_DN and OLTS_ATTRSTORE)

RAID

The information on balancing tablespaces is given in terms of separating Oracle Internet Directory tablespaces onto different logical drives. This assumes that a 'logical drive' is manifested on a separate disk or set of disks from other 'logical drives', and thus represents a division among disks for I/O. (Two logical drives on the same physical disk media do not really provide the same combined I/O throughput of two logical drives located on different physical media.) If a logical drive can be manifest on a striped or RAID disk subsystem, then this may increase the I/O capacity of that logical drive. However, the tablespace locations considered earlier remain applicable when considering, for instance, different logical drives of a volume manager.

Database Tuning

This section describes the other tunable parameters available to an Oracle Internet Directory installation.

The following table gives a quick overview of the recommended values of RDBMS parameters for various client loads. These parameters are configurable in the initialization parameter file.

Parameters	500 Concurrent LDAP Clients	1000 Concurrent LDAP Clients	1500 Concurrent LDAP Clients	2000 Concurrent LDAP Clients
Open_cursors	100	100	100	100
Sessions	225	600	800	1200
Database_block_buffers	200 to 250 MB	200 to 250 MB	200 to 250 MB	200 to 250 MB
Database_block_size	8192	8192	8192	8192
Shared_pool_size	30 to 40 MB	30 to 40 MB	30 to 40 MB	30 to 40 MB
Processes	400	800	1000	1500

This section describes each of the RDBMS tunable parameters in more detail. It contains these topics:

- Required Parameter
- Parameters Dependent on Oracle Internet Directory Server Configuration
- SGA Parameters Dependent on Hardware Resources

Required Parameter

Configure the OPEN_CURSORS parameter as follows:

```
OPEN_CURSORS=100
```

The Oracle9*i* default of 50 or so is too small to accommodate Oracle Internet Directory server cursor cache. Note that this value is not dependent on other Oracle Internet Directory server parameters, such as # SERVERS and # WORKERS. The value of 100 is sufficient for any size DIT.

Parameters Dependent on Oracle Internet Directory Server Configuration

Configure the SESSIONS parameter as follows:

```
PROCESSES = (# OID server processes per instance) x
            (# DB Connections per server + 1) x
            (# of OID instances) + 20
SESSIONS = 1.1 * PROCESSES + 5
```

Each Oracle Internet Directory server process requires a number of concurrent database connections equal to the number of worker threads configured for that server plus one. The total number of concurrent database connections allowed must therefore include this number per server, per instance. The additional 20 connections added to the parameter value accounts for the Oracle background processes plus other Oracle Internet Directory processes such as OID Monitor, OID Control, Oracle directory replication server, and bulk tools.

Using Shared Server Process

Depending on the total number of concurrent database connections required, and as determined by the setting for the SESSIONS parameter, enabling shared server process may help balance overall system load better. If the total number of concurrent database connections required is over 300, then configure the shared server. One shared server should be configured for every 10 database connections required.

> **Note:** The number of required concurrent database connections depends on the hardware selected. See *Oracle Net Services Administrator's Guide* and *Oracle9i Database Administrator's Guide* for further information about the shared server configuration.

SGA Parameters Dependent on Hardware Resources

The main parameters that contribute to the SGA are discussed in "Memory Tuning" on page 19-7. The following are a few more parameters that may be tuned:

- Sort area

  Set to 262144 (256k) to ensure sufficient sort area available to prevent on-disk sorts.

- Redo Log Buffers

  Set to 32768 (32k) as an initial estimate. If log write performance becomes a performance problem, use a large enough value to make sure (redo log space requests / redo entries) > 1/5000 to prevent the LGWR process from falling behind. This overall has little size effect on the variable SGA size, so making this a little bit too large should not be a problem.

Performance Troubleshooting

This section gives some quick pointers for common performance related problems.

If LDAP search performance is poor, make sure that:

- The attributes on which the search is being made are indexed
- Schema associated with the `ODS` user is `ANALYZED`

  For searches involving multiple filter operands, make sure that the order in which they are given goes from the 'most specific' to the 'least specific'. For example, `&(l=Chicago)(state=Illinois)(c=US)` is better than `&(c=US)(state=Illinois)(l=Chicago)`.

If LDAP add/modify performance is poor, make sure that:

- There are enough redo-log files in the database
- The undo tablespace in the database is large enough
- The schema associated with the `ODS` user is `ANALYZED`

Part VI

The Directory and Clusters

This part contains these chapters:

20

Managing Failover in Cluster Configurations

This chapter contains these topics:

- Introduction
- Configuring Failover in a Clustered Environment
- How Failover Works in a Clustered Environment

Introduction

Oracle Internet Directory release 3.0.1 enables you to increase high availability by using logical hosts—as opposed to physical hosts—in clustered environments.

A logical host consists of one or more disk groups, and pairs of host names and IP addresses. It is mapped to a physical host in the cluster. This physical host impersonates the host name and IP address of the logical host.

In this paradigm, the directory server binds to the logical host, rather than the physical host. It maintains this connection even if the logical host fails over to a new physical host.

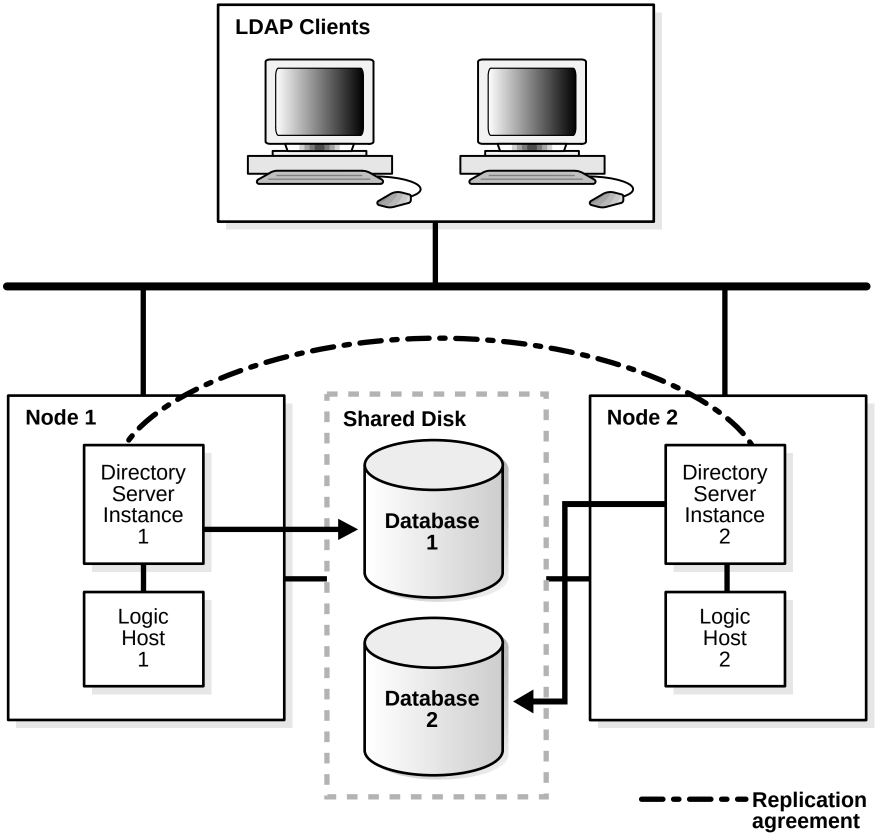
A client connects to the directory server by using the logical host name and address of the server. If the logical host fails over to a new physical host, then that failover is transparent to the client.

A logical host can reside on two or more cluster nodes that have physical access to its disk storage. A cluster can typically support any number of logical hosts, and a physical server or cluster node can impersonate more than one logical host.

This failover mechanism also supports replicated environments.

Figure 20–1 shows a sample Oracle Internet Directory configuration on a hardware cluster.

Figure 20–1 Oracle Internet Directory Configuration in a Two-Node Cluster



In this configuration:

- Physical Node 1 masters Logical Host 1
- Physical Node 2 masters Logical Host 2
- Directory Server Instance 1, consisting of one or more directory server instances, runs on Logical Host 1
- Directory Server Instance 2, consisting of one or more directory server instances, runs on Logical Host 2
- Both directory server instances have their respective directory data stores—Oracle databases—on the shared disk
- Directory Server Instance 1 and Directory Server Instance 2 are in a replication agreement

Clients connect to Directory Server Instance 1 by using the host name and address of Logical Host 1. Similarly, clients connect to Directory Server Instance 2 by using the host name and address of Logical Host 2.

Configuring Failover in a Clustered Environment

This section tells you how to configure failover in a clustered environment.

Note: At the end of Oracle Internet Directory installation, a directory server instance and the OID Monitor are started by default. To run Oracle Internet Directory on a logical host, you must stop the directory server instance and the OID Monitor, then restart them by using either of the optional flags `-host` or `-h`. Do this before any updates are made to the directory. This way, you ensure that the directory server uses the logical host name in change log generation.

It contains these topics:

- Step 1: Start OID Monitor
- Step 2: Start a Directory Server or Directory Replication Server by Using the OID Control Utility
- Step 3: Stop, then Restart, the Directory Server and OID Monitor

Step 1: Start OID Monitor

When you start OID Monitor, use the optional host argument, and set it to the logical host name. In the following example, OID Monitor connects to the directory store, my_net_service and monitors the directory server instances on the logical host, my_host.:

```
oidmon [connect=my_net_service] host=my_host
```

Step 2: Start a Directory Server or Directory Replication Server by Using the OID Control Utility

When you start the directory server by using the OID Control utility, use either of the optional flags -host or -h, and set it to the logical host name. In the following example, the OID Control utility directs the OID Monitor to start the directory server instance on the logical host, my_host.

```
oidctl connect=my_net_service server=oidldapd instance=1 flags=”-h my_host”
start
```

Similarly, when you start a directory replication server by using the OID Control utility, use either of the optional flags -host or -h, and set it to the logical host name. In the following example, the OID Control utility directs the OID Monitor to start the directory replication server on the logical host, my_host.

```
oidctl connect=my_net_service server==oidrepld instance=1 flags=”-h my_host”
start
```

Note: The replication agreement should use logical host names rather than physical host names for specifying the host names.

Step 3: Stop, then Restart, the Directory Server and OID Monitor

To run Oracle Internet Directory on a logical host, stop the directory server instance and the OID Monitor, then restart them by using either of the optional flags `-host` or `-h`. Do this before any updates are made to the directory. This way, you ensure that the directory server uses the logical host name in change log generation.

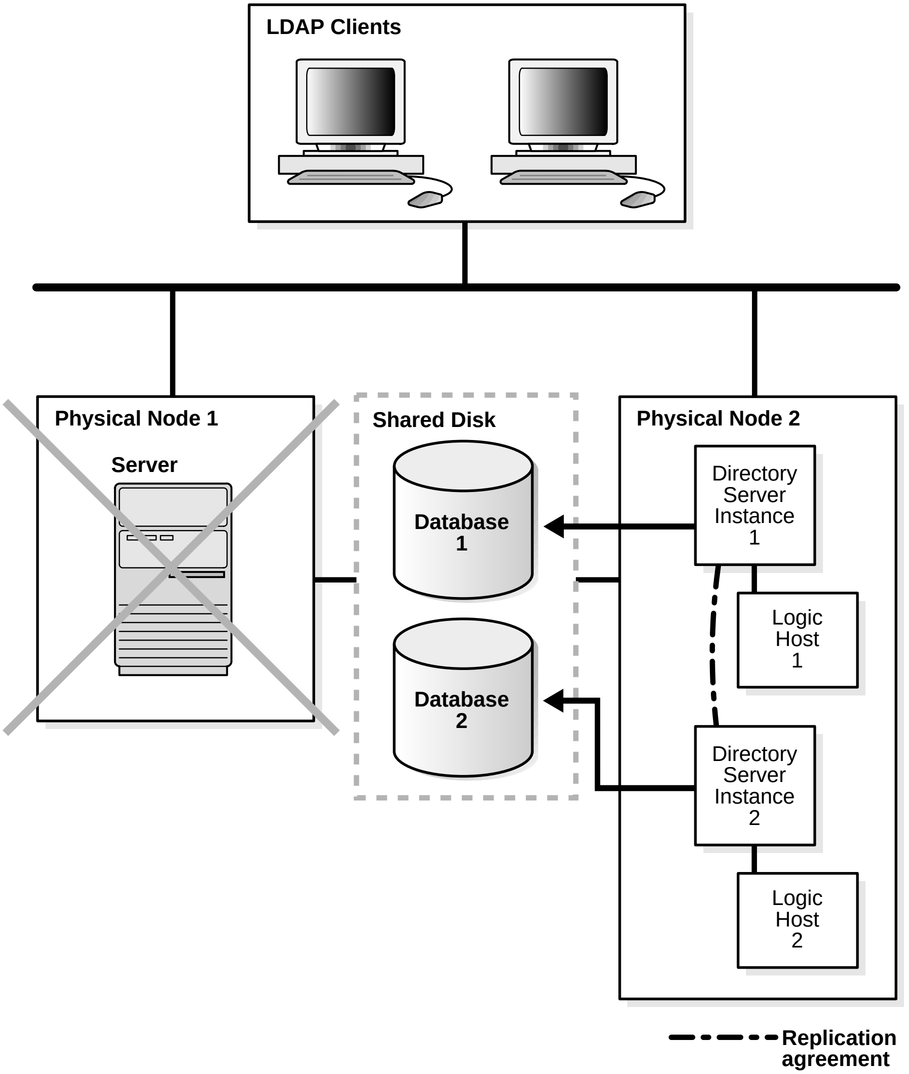
See Also:

- "Stopping an Oracle Directory Server Instance" on page 4-5
- "Stopping the OID Monitor" on page 4-3
- "Starting an Oracle Directory Server Instance" on page 4-4
- "Starting the OID Monitor" on page 4-2

How Failover Works in a Clustered Environment

Figure 20–2 shows a scenario in which a failover has occurred and the directory server has been restarted.

Figure 20–2 Oracle Internet Directory Nodes After Failover



In Figure 20–2, Physical Node 1 fails. At that point, Logical Host 1 fails over to be mastered by Physical Node 2. After this has finished, Directory Server Instance 1 needs to be restarted—that is, OID Monitor needs to be restarted with Logical Host 1 specified as the host name.

This failover of Directory Server Instance 1 is transparent to the LDAP clients connecting to Directory Server Instance 1. These clients continue to connect to Directory Server Instance 1 by using the host name and address of Logical Host 1.

After the failover, Directory Server Instance 1 continues to use the host name of Logical Host 1 in the change log generation. The replication agreement between Directory Server Instance 1 and Directory Server Instance 2 continues as before the failover.

21

Managing Directory Failover in an Oracle9*i* Real Application Clusters Environment

Oracle9*i* Real Application Clusters is a computing environment that harnesses the processing power of multiple, interconnected computers. Along with a collection of hardware, called a cluster, it unites the processing power of each component to become a single, robust computing environment. A cluster comprises two or more computers, also called nodes.

This chapter discusses the ways you can run Oracle Internet Directory in an Oracle Real Application Clusters system. It contains these topics:

- The Oracle Directory Server in an Oracle9i Real Application Clusters Environment
- The Oracle Directory Replication Server in an Oracle9i Real Application Clusters Environment

Terminology

Node

A computer where an instance resides. It can be part of a Massively Parallel Computing Infrastructure where it shares disk storage with other nodes. In most cases, a node has its own copy of the operating system.

Cluster

A set of instances, each typically running on different nodes, that coordinate with one another when accessing the shared database on the disk.

Cluster Manager

An operating system dependent component that discovers and tracks the membership state of nodes by providing a common view of cluster membership across the cluster.

Transparent Application Failover (TAF)

A runtime failover for high-availability environments, such as Oracle Real Application Clusters and Oracle Fail Safe, that refers to the failover and re-establishment of application-to-service connections. It allows client applications to automatically reconnect to the database if the connection fails, and optionally resume a SELECT statement that was in progress. This reconnect happens automatically from within the Oracle Call Interface (OCI)

The client notices no connection loss as long as there is one instance left serving the application.

Connect-time failover

Failover method in which a client connect request is forwarded to a another listener if the first listener is not responding. It is enabled by service registration, because the listener knows if an instance is running before attempting a connection.

The Oracle Directory Server in an Oracle9*i* Real Application Clusters Environment

You can run a directory server on a node that is different from the one running the cluster database. The computer on which the directory server runs may be part of the cluster.

This section contains these topics:

- Oracle Internet Directory with Basic High Availability Configuration
- Oracle Internet Directory with Default N-Node Configuration

Oracle Internet Directory with Basic High Availability Configuration

In this case, a single directory server connects to two or more Real Application Clusters instances, each running on different nodes. This scenario is easy to configure and, on the node where the primary instance is running, it provides greater resilience after either a hardware or software failure.

Figure 21–1 shows the setup in detail.

Figure 21–1 Oracle Internet Directory with Basic High Availability Configuration

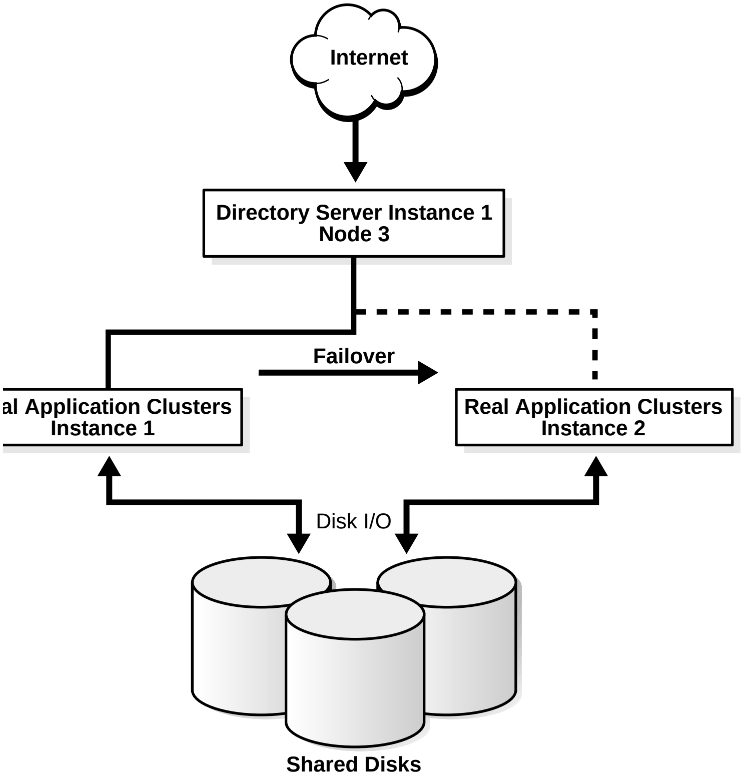


Figure 21–1 shows a three-node cluster. Real Application Clusters Instance 1 runs on Node 1. Real Application Clusters Instance 2 runs on Node 2. The directory server instance runs on Node 3.

Normally, the directory server instance communicates with the Real Application Clusters Instance on Node 1, which is the primary instance. However, in the event of either a hardware or software failure on a Node 1, Oracle Net Services can redirect database requests to the Real Application Clusters instance on Node 2, the secondary instance.

To specify the primary instance, in the initialization file, set the ACTIVE_INSTANCE_COUNT parameter to 1 for both instances. The instance you start first becomes the primary instance.

The primary instance can accept connections from its local listener, as well as from the secondary instance listener. A secondary instance registers with its local listener as a secondary instance, and like the primary instance, its ACTIVE_INSTANCE_COUNT parameter is set to 1. If the primary instance fails, then the secondary instance assumes the primary role and registers with its listeners. When the failed instance can once again start, it does so as the secondary instance. If you have failover configured, then directory server connections to the failed primary instance fail over to the secondary instance.

The following is an example of a tnsnames.ora file configured for a connect-time failover. In this example, the LOAD_BALANCE must be set to OFF.

```
MY_CLUSTER =
  (DESCRIPTION =
    (LOAD_BALANCE = OFF)
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = my_cluster.my_company.com))
  )
MY_CLUSTER_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_1)
    )
  )
MY_CLUSTER_2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = dlsun722)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_2)
    )
  )
```

The following is an example of a `listener.ora` file configured for a connect-time failover.

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = my_host)(PORT = 1521))
      )
    )
  )
```

The following is an example of a `tnsnames.ora` file configured for a transparent application failover (TAF).

```
MY_CLUSTER =
  (DESCRIPTION =
    (FAILOVER = ON)
    (LOAD_BALANCE = OFF)
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = my_cluster.my_company.com)
      (FAILOVER_MODE = (TYPE = SELECT) (METHOD = PRECONNECT)
                       (BACKUP = ops1))
    )
  )
MY_CLUSTER_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = ops1)
    )
  )
MY_CLUSTER_2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_2)
    )
  )
```

The following is an example of a `listener.ora` file configured for a transparent application failover (TAF)

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
      )
    )
  )
```

Note: Depending on the state of the directory server when the database failure occurs, Oracle Internet Directory may not successfully manage the transparent application failover. In this case, Oracle Internet Directory logs "ORA-25402—Transaction must rollback" in the log file and re-establishes a new database connection against a live database instance. The client may receive the error message "DSA unwilling to perform." When this happens, the client can simply reissue the request to the directory server.

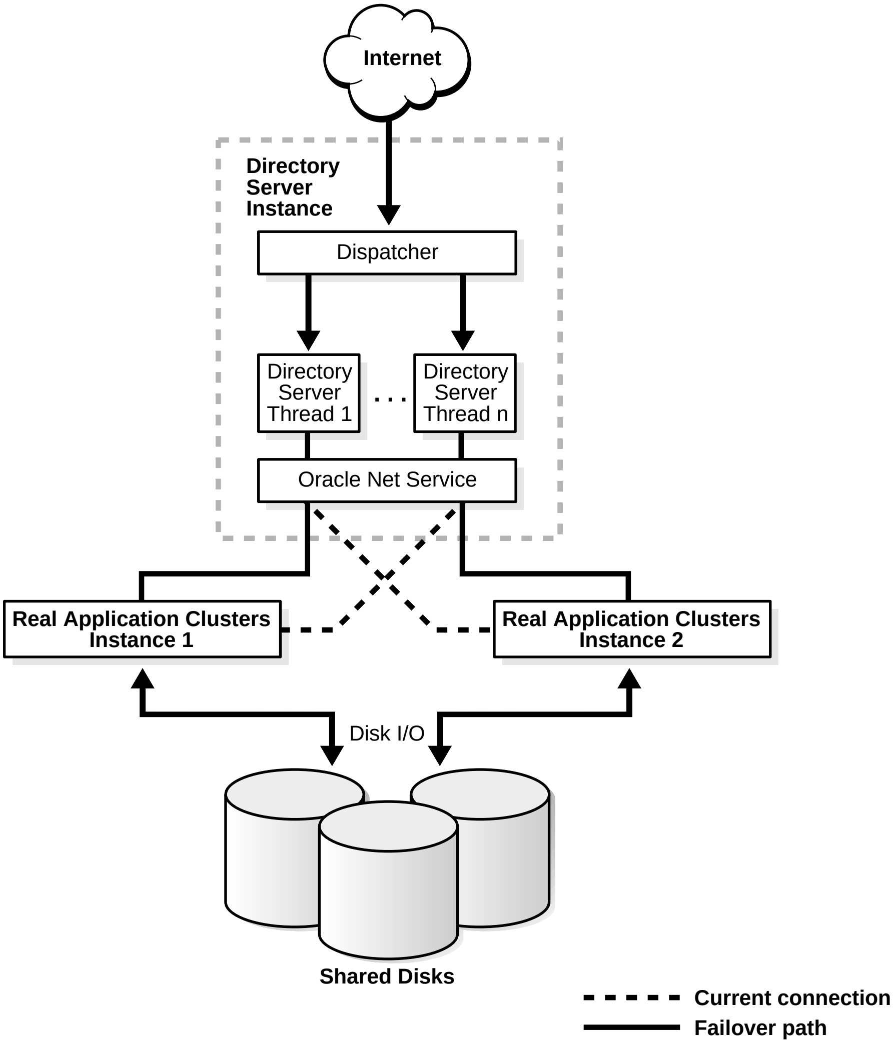
Oracle Internet Directory with Default N-Node Configuration

In this case, there are multiple directory server threads connecting to two or more Real Application Clusters instances on different nodes. To achieve this, you can set the `LOAD_BALANCE` parameter of Oracle Net Services to `ON`.

Figure 21–2 shows a three-node cluster. Real Application Clusters Instance 1 runs on Node 1. Real Application Clusters Instance 2 runs on Node 2. A directory server instance with Directory Server Thread #1 and Directory Server Thread #2 runs on Node #3.

> **See Also:** *Oracle Net Services Administrator's Guide* for instructions on setting the LOAD_BALANCE parameter

Figure 21–2 Single Directory Server Instance on One Node and Multiple Real Application Clusters Instances


Internet
Directory Server Instance
Dispatcher
Directory Server Thread 1
. . .
Directory Server Thread n
Oracle Net Service
Real Application Clusters Instance 1
Real Application Clusters Instance 2
Disk I/O
Shared Disks
Current connection
Failover path

Depending on how the Oracle Net Services routes the LDAP request, when all nodes in Figure 21–2 are running, directory server thread 1 may connect to Real Application Clusters Instance 1, and directory server thread 2 may connect to Real Application Clusters Instance 2. Incoming LDAP requests to the directory server are distributed in round-robin fashion to both directory database connections. If there is a hardware or software failure on node 1, then directory server thread 1 reconnects to Real Application Clusters Instance 2 by using connection time failover or Oracle Net transparent application failover.

The scenario in this example provides higher availability and scalability. If the database or database host fails, then it provides resilience by using connection time failover or Oracle Net transparent application failover. Moreover, it provides higher throughput for complicated LDAP subtree searches. However, it may induce a cluster pinging problem if there is a large volume of updates, and it does not recover in the event of failure on the node running the directory server.

To configure your system for this scenario, examine the following examples of the various configuration files.

The following example shows a `tnsnames.ora` file configured for a connect-time failover.

```
MY_CLUSTER =
  (DESCRIPTION =
    (LOAD_BALANCE = ON)
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = my_cluster.my_company.com))
  )
MY_CLUSTER_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_1)
    )
  )
MY_CLUSTER_2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_2)
    )
  )
```

The following example shows a `listener.ora` file configured for connect time failover.

```
# LISTENER.ORA Network Configuration
# File: D:\oracle\ora81\database\opstemp\atlnt10i\network\admin\listener.ora
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = dlsun722)(PORT = 1521))
      )
    )
  )
```

The following two examples show two `tnsnames.ora` files, one for `my_host_1` and the other for `my_host_2`, configured for a transparent application failover (TAF).

The `tnsnames.ora` on `my_host_1`:

```
MY_CLUSTER =
  (DESCRIPTION =
    (FAILOVER = ON)
    (LOAD_BALANCE = ON)
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = my_cluster.my_company.com)
      (FAILOVER_MODE = (TYPE = SELECT) (METHOD = PRECONNECT)
                       (BACKUP = my_host_1))
    )
  )
```

```
MY_CLUSTER_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_company_1)
    )
  )
OPS2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_company_2)
    )
  )
```

The `tnsnames.ora` on `my_host_2`:

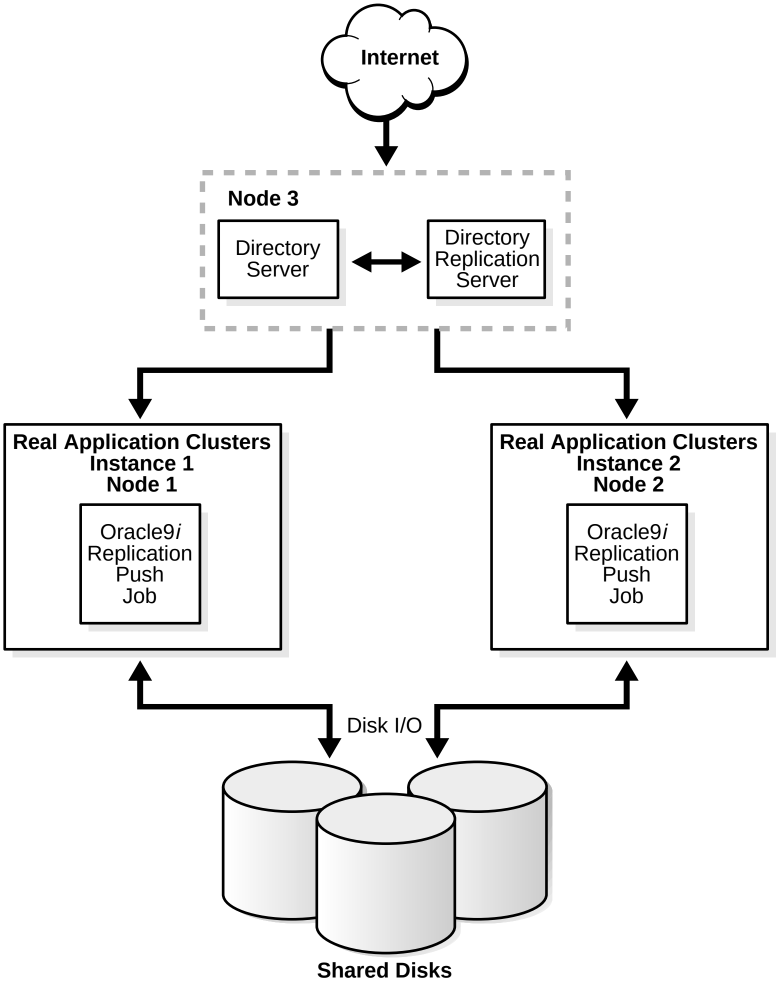
```
MY_CLUSTER =
  (DESCRIPTION =
    (FAILOVER = ON)
    (LOAD_BALANCE = ON)
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = my_cluster.my_company.com)
      (FAILOVER_MODE = (TYPE = SELECT) (METHOD = PRECONNECT)
                       (BACKUP = my_company_2))
    )
  )
MY_CLUSTER_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_1)
    )
  )
```

```
MY_CLUSTER_2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_2)
    )
  )
```

The Oracle Directory Replication Server in an Oracle9*i* Real Application Clusters Environment

Figure 21–3 shows a possible Oracle directory replication server in an Oracle Real Application Clusters environment.

Figure 21–3 Directory Replication Server in an Oracle Real Application Clusters Environment


Internet
Node 3
Directory Server
Directory Replication Server
Real Application Clusters Instance 1 Node 1
Oracle9*i* Replication Push Job
Real Application Clusters Instance 2 Node 2
Oracle9*i* Replication Push Job
Disk I/O
Shared Disks

There are three nodes in this configuration. A directory server instance runs on Node 3, and Real Application Clusters instances run on Node 1 and Node 2. When all nodes are running, the directory replication server connects to the directory server instance, and the Oracle9*i* Replication push jobs are running on both Real Application Clusters instances. If there is any hardware failure on Node 3, the directory replication server on Node 2 restarts and connects to directory server instance 2. If any hardware failure happens on Node 2, then, after cluster reconfiguration, the Oracle9*i* Replication push job continues on Real Application Clusters instance 1.

This scenario provides resilience in the event of database or database host failure for replication data transfer, that is, an Oracle9*i* Replication push job. It also provides resilience in the event of a directory server instance or host failure, or failure for the directory replication server.

Part VII

The Oracle Directory Integration Platform

This part explains the concepts, architecture, and components of the Oracle Directory Integration platform, and tells you how to configure and use it to synchronize multiple directories with Oracle Internet Directory. It contains these chapters:

22

About the Oracle Directory Integration Platform

This chapter introduces the Oracle Directory Integration platform, its components, architecture, and administration tools.

This chapter contains these topics:

What Is the Oracle Directory Integration Platform?

The Oracle Directory Integration platform enables you to synchronize various directories with Oracle Internet Directory. It also makes it easier for third party metadirectory vendors and developers to develop and deploy their own connectivity agents.

This section contains these topics:

- About Metadirectories
- About the Oracle Directory Integration Platform Environment

About Metadirectories

Enterprises today often deploy multiple directories to store information for applications such as ERP systems, database applications, messaging systems, and Network Operating Systems (NOS). Managing so many different directories has many drawbacks, including:

- Increased cost—Multiple administrators must maintain essentially the same information in many different places.
- Inconsistent data—Updated information in one directory is not available to all the other directories.

A metadirectory solves these problems by synchronizing information between all enterprise directories, forming one virtual directory. It centralizes administration, thereby reducing administrative costs. It ensures that data is consistent and up-to-date across the enterprise.

For example, in a metadirectory environment, you can create a global directory entry for each employee. You can populate this entry with data from various synchronized directories—for example, Human Resources applications, messaging systems, or NOS databases. Users can then access this global entry, knowing that the data it contains is up-to-date and synchronized with each **connected directory**.

You can also ensure that the synchronization process respects all existing data ownership policies. For example, you can grant to only the Human Resources department the privilege to change the value of an employee’s salary attribute.

About the Oracle Directory Integration Platform Environment

In an Oracle Directory Integration platform environment, each connected directory synchronizes with Oracle Internet Directory, which serves as the central directory. This provides:

- Consistent, up-to-date information for both users and applications
- A single point of access to all directory data through standards-based clients—for example, Web browsers or email clients
- A central point of administration of all enterprise directories

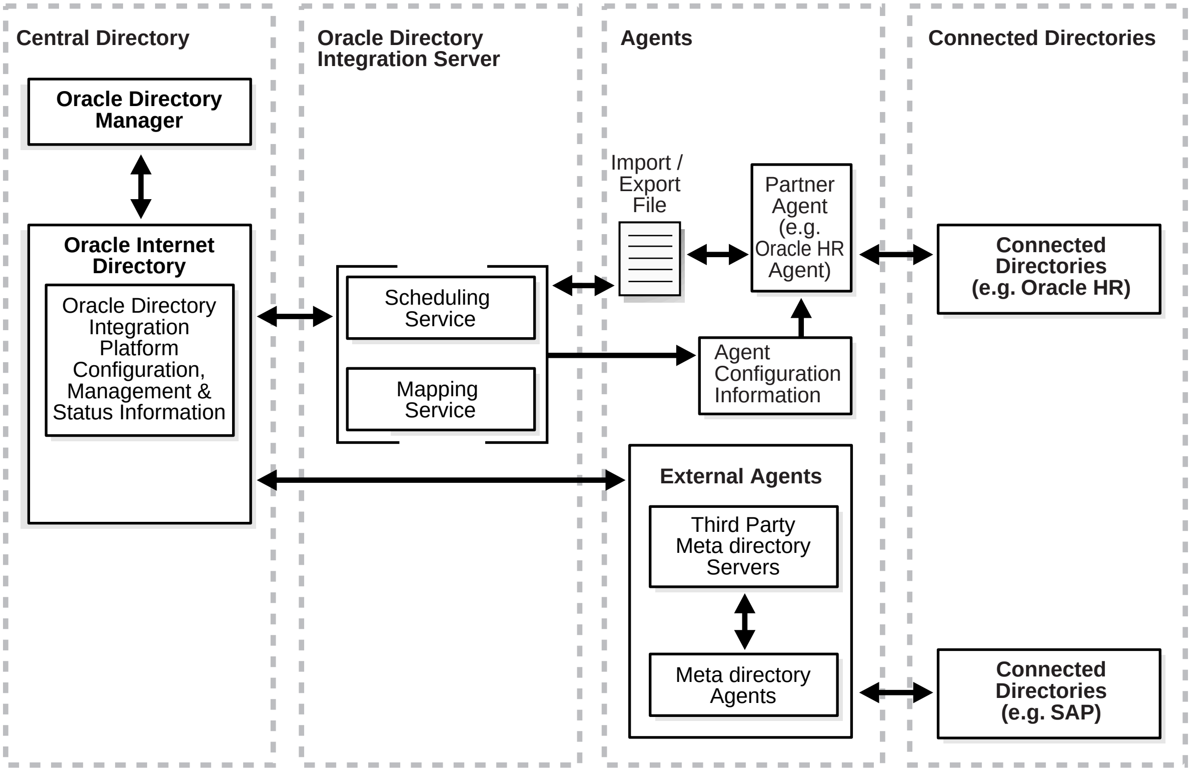
Oracle Directory Integration platform enables you to:

- Import data from connected directories into Oracle Internet Directory, either all at once or incrementally
- Export data from Oracle Internet Directory into connected directories, either all at once or incrementally
- Synchronize all or part of the data in a connected directory with Oracle Internet Directory. For example, you can decide to synchronize the user name attributes, but not the salary attributes, for enterprise employees.

Architecture

Figure 1–1 shows the architecture of the Oracle Directory Integration platform:

Figure 22–1 Oracle Directory Integration Platform Architecture


Central Directory
Oracle Directory Manager
Oracle Internet Directory
Oracle Directory Integration Platform Configuration, Management & Status Information
Oracle Directory Integration Server
Scheduling Service
Mapping Service
Agents
Import / Export File
Partner Agent (e.g. Oracle HR Agent)
Agent Configuration Information
External Agents
Third Party Meta directory Servers
Meta directory Agents
Connected Directories
Connected Directories (e.g. Oracle HR)
Connected Directories (e.g. SAP)

The following sections describe each component and its relation to the rest of the Oracle Directory Integration platform.

Components of the Oracle Directory Integration Platform Architecture

This section contains these topics:

- Oracle Internet Directory
- Connected Directories
- Oracle Directory Integration Server
- Directory Integration Agents
- Import and Export Files
- Directory Integration Toolkit

Oracle Internet Directory

Oracle Internet Directory release 3.0.1 is an LDAP v3-compliant directory server that uses Oracle9*i* as a data store. In the Oracle Directory Integration platform, it is the central directory for all information, the directory against which all other directories are synchronized.

This synchronization is bidirectional: Changes in Oracle Internet Directory are exported to connected directories, and changes in connected directories are imported into Oracle Internet Directory.

In an Oracle Internet Directory environment with multiple nodes, Oracle Internet Directory synchronizes the directory servers by using its own replication capabilities instead of the platform.

Connected Directories

In the Oracle Directory Integration platform environment, connected directories are those other than Oracle Internet Directory, the central directory. They could include, for example, relational databases, Oracle HR, Microsoft Exchange, or Lotus Notes.

Oracle Directory Integration Server

The Oracle directory integration server, a multithreaded daemon server process, is the central component of Oracle Directory Integration platform. It performs:

- Scheduling—Running a **directory integration agent** at a time you specify
- Mapping—Executing rules for converting data between connected directories and Oracle Internet Directory
- Error handling

You can run multiple servers, each on a different computer. You can also run multiple instances of directory integration server on the same computer at the same time. Each instance has a **configuration set entry** listing the agents the Oracle directory integration server instance is to run.

See Also:

- "Managing Server Configuration Set Entries" on page 6-2
- "Directory Integration Agents" on page 22-6 for a discussion of agents and directory integration profiles

Directory Integration Agents

A directory integration agent is a program that synchronizes data between Oracle Internet Directory and connected directories. When it synchronizes the data, it does one or more of the following:

- Exports changes out of Oracle Internet Directory
- Imports changes into a connected directory
- Exports changes out of a connected directory
- Imports changes into Oracle Internet Directory

Depending on how it is deployed in the Oracle Directory Integration platform,an agent is known as either a **partner agent**or an **external agent**.

Partner agents run under the control of the Oracle directory integration server—that is, the Oracle directory integration server performs scheduling, data mapping, and error handling for them. Before deploying a partner agent, you register it in Oracle Internet Directory. This registration involves creating a **directory integration profile** in the directory. To create the profile, you can use either Oracle Directory Manager or command-line tools.

A partner agent uses either an **import file** or an **export file** to exchange data between a connected directory and Oracle Internet Directory. At execution time, they may use additional agent configuration information stored in Oracle Internet Directory.

Unlike partner agents, external agents are independent of the Oracle directory integration server—that is, the Oracle directory integration server performs neither scheduling nor data mapping for them. You do not need to register external agents with Oracle Internet Directory.

Typically, you use external agents when a third party metadirectory solution is integrated with the platform. In this case, the third party metadirectory solution uses its own metadirectory engine to perform mapping and scheduling.

See Also:

- "Directory Integration Profiles" on page 22-8 for a fuller explanation of the directory integration profile
- "Oracle Directory Integration Server" on page 22-6 for a fuller explanation of the role of the the Oracle directory integration server

Import and Export Files

These files store data extracted from either a connected directory or Oracle Internet Directory. The platform uses them to exchange data between Oracle Internet Directory and connected directories.

Import data files are those to which changes in connected directories are written. Export data files are those to which changes in Oracle Internet Directory are written.

Directory Integration Toolkit

The directory integration toolkit allows third party metadirectory vendors and developers to integrate their metadirectory solutions with the Oracle Directory Integration platform environment The toolkit consists of:

- Interfaces for accessing changes in Oracle Internet Directory by clients:
  - IETF standard change log interface
  - Oracle proprietary change log interface

- Interfaces to register directory integration agents into Oracle Internet Directory:
  - Oracle Directory Manager
  - Command-line tools to add and modify data by using an LDIF file configuration
- An interface for scheduling agents by using the Oracle directory integration server
- An interface for data mappings by using the Oracle directory integration server
- Tools and procedures for bootstrapping connected directories into the Oracle Directory Integration platform environment. These enable you to:
  - Bulk import data from LDIF files
  - Bulk export Oracle Internet Directory data into LDIF files

Directory Integration Profiles

A directory integration profile contains configuration information required for synchronization—for example, the name and type of an agent, how and when to invoke it, the mapping information required for synchronization, and status information. There must be a directory integration profile for each partner agent.

The directory integration profile is managed in the directory. You create it by using either Oracle Directory Manager or the command-line tools.

This section discusses two elements of the directory integration profile. It contains these topics:

- Agent Configuration Information
- Attribute Mapping Rules

Agent Configuration Information

An agent may need some configuration information at runtime for performing various operations. For example, to make it easier for users to specify which connected directory attributes are to be synchronized with Oracle Internet Directory, you may want an agent to store a list of these attributes as part of its configuration information. This kind of information is called agent configuration information.

You can store agent configuration information wherever and however you want. However, the Oracle Directory Integration platform enables you to store it as a binary attribute, called orclIPAgentConfigInfo, in the directory integration profile. The Oracle directory integration server passes this information as a temporary file to the agent at the time of the agent's invocation.

Agent configuration information is optional. If an agent does not require such information, then the corresponding attribute in the Oracle Directory Integration platform profile is left empty.

See Also: Chapter 23, "Managing Directory Integration Agents and Profiles"

Attribute Mapping Rules

Mapping rules govern the conversion of attributes between a connected directory and Oracle Internet Directory. There is one set of mapping rules for each connected directory. This set is stored as a binary value in an attribute called orclODIPAttributeMappingRules in the integration profile in Oracle Internet Directory.

The directory integration server uses these rules to map attributes, as necessary, when generating an export file or interpreting an import file. When the directory integration server imports changes into Oracle Internet Directory, it converts the connected directory change records into LDAP change records, following the mapping rules specified in the integration profile. Similarly, when the directory integration server exports changes from Oracle Internet Directory, it converts the Oracle Internet Directory change records into connected directory change records, following the mapping rules specified in the integration profile.

It supports both one-to-many and many-to-one mapping.

One-to-many mapping

The directory integration server can map one attribute in a connected directory to many attributes in Oracle Internet Directory. For example, it can map an attribute in the connected directory—`Address:123 Main Street/MyTown, MyState 12345`—to both of the two LDAP attributes `homeAddress` and `postalAddress`.

Many-to-one mapping

The directory integration server can map multiple attributes in a connected directory to one attribute in Oracle Internet Directory. For example, suppose that the Human Resources directory represents Anne Smith by using two attributes: `firstname=Anne` and `lastname=Smith`. The directory integration server can map these two attributes to one attribute in Oracle Internet Directory: `cn=Anne Smith`.

Administration and Monitoring Tools

This section contains these topics:

- Oracle Directory Manager
- OID Control and OID Monitor

Oracle Directory Manager

Oracle Directory Manager, a Java-based graphical user interface tool, enables you to administer the Oracle Directory Integration platform. Specifically, it enables you to:

- Create, modify, and delete directory integration profiles
- Check the status of agents
- Check the status of all the Oracle directory integration server instances

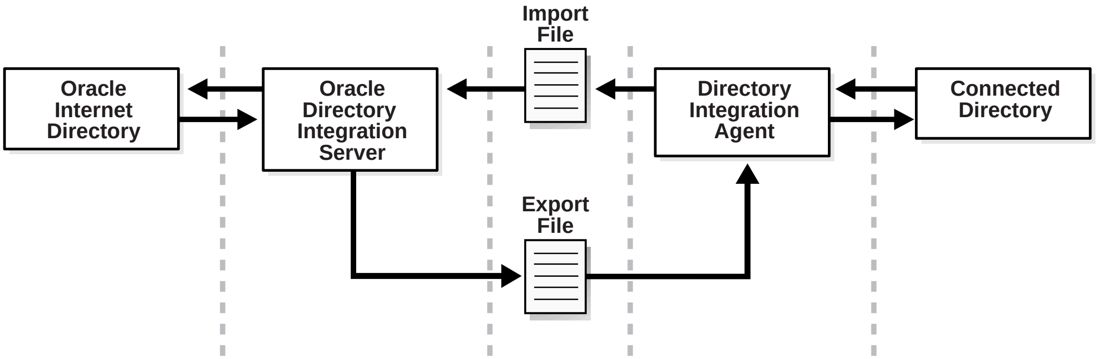
OID Control and OID Monitor

OID Control and OID Monitor enable you to start, stop, and monitor the Oracle directory integration server.

In Oracle Internet Directory release 3.0.1, you can use OID Control and OID Monitor to control the directory integration server only on a host containing Oracle Internet Directory server installations. If Oracle Internet Directory installation is client-only, then the OID Control utility and OID Monitor are not installed. In this case, start the Oracle directory integration server manually. In this configuration you can still use Oracle Directory Manager to learn the status of the Oracle directory integration server.

How the Oracle Directory Integration Platform Works

This diagram shows the directions in which information flows in an import operation and in an export operation.


Import File
Oracle Internet Directory
Oracle Directory Integration Server
Directory Integration Agent
Connected Directory
Export File

To export changes from Oracle Internet Directory to a connected directory, the Oracle directory integration server first retrieves from Oracle Internet Directory any change records it has not earlier retrieved for the connected directory. It writes these records to an export file, then starts the agent. The agent:

1. Reads the export file
2. Performs attribute mappings
3. Updates the information in the connected directory

To keep track of changes already applied by directory integration agents, Oracle Internet Directory maintains a change log. It does not purge change log information until the appropriate directories have consumed the changes.

To import changes into Oracle Internet Directory, the Oracle directory integration server first starts the agent at the specified time. The agent extracts change records from the connected directory and writes them to an import file. The directory integration server:

1. Reads this import file
2. If necessary, maps the attributes
3. Updates the entry in Oracle Internet Directory with the changes from the connected directory

A Scenario: Deploying Oracle Human Resources Agent

release 3.0.1 of Oracle Directory Integration platform includes an agent for Oracle HR.

Although an enterprise deploying Oracle Internet Directory may store employee data in Oracle Internet Directory, the Human Resources department typically controls that data. In an enterprise deploying both Oracle Human Resources and Oracle Internet Directory, Oracle Directory Integration platform synchronizes the employee data from Oracle Human Resources into Oracle Internet Directory.

The Oracle Human Resources agent extracts changes from Oracle Human Resources and places them in an import file. The Oracle directory integration server extracts those changes from the file and imports them into Oracle Internet Directory. This enables Oracle Human Resources to be the source of truth for employee information. All LDAP-enabled applications can then access up-to-date employee data from Oracle Internet Directory.

Note: Oracle Internet Directory release 3.0.1 does not allow changes in Oracle Internet Directory to be exported to Oracle HR.

23

Managing Directory Integration Agents and Profiles

This chapter discusses directory integration agents and the operations they perform in the Oracle Directory Integration platform. It explains how to manage partner agents by using either Oracle Directory Manager of command-line tools. It contains these topics:

- About Directory Integration Agents
- Managing Partner Agents

About Directory Integration Agents

This section contains these topics:

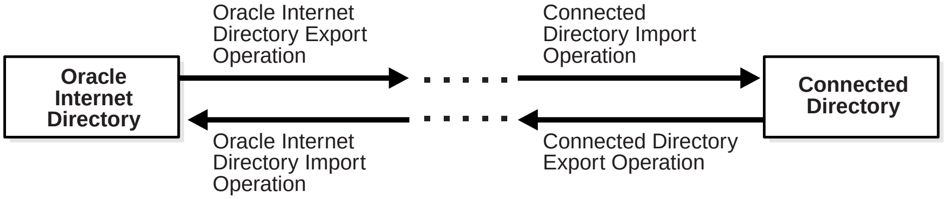
- Import and Export Operations
- Synchronization Scenarios
- Types of Agents
- Change Log Interfaces
- Registration of Partner Agents into Oracle Directory Integration Platform
- Agent Configuration Information
- Mapping Rules
- File Naming Conventions
- Location of Files

Import and Export Operations

Agents are programs that perform one or more of the following operations, each of which is discussed in this section:

- Oracle Internet Directory Export Operation—For exporting changes out of Oracle Internet Directory
- Connected Directory Import Operation—For importing changes into a connected directory
- Connected Directory Export Operation—For exporting changes out of a connected directory
- Oracle Internet Directory Import Operation—For importing changes into Oracle Internet Directory

The following diagram shows the direction in which the data flows in each operation between Oracle Internet Directory and a connected directory. The remainder of this section describes each operation.


Oracle Internet Directory Export Operation
Connected Directory Import Operation
Oracle Internet Directory
Connected Directory
Oracle Internet Directory Import Operation
Connected Directory Export Operation

Oracle Internet Directory Export Operation

An Oracle Internet Directory export operation consists of:

- Reading all the required changes in Oracle Internet Directory from the time of the last retrieval
- Writing those changes into an export file. If the export file needs to be written in a format other than LDIF, then this task includes converting the LDAP change records into the connected directory change records.
- Keeping track of the last time the changes were extracted from Oracle Internet Directory. This includes ensuring that the changes are retrieved before they are purged from Oracle Internet Directory.
- Ensuring that only those changes required by the connected directory are written to the export file

Connected Directory Import Operation

A connected directory import operation consists of:

- Reading all the Oracle Internet Directory changes that the exporting agent wrote into the export file. If the export file is written in LDIF, then this task includes converting the LDAP change records into the connected directory change records.
- Updating the change records in the connected directory
- After all the changes are updated in the connected directory, moving the export file to an archive directory

Connected Directory Export Operation

A connected directory export operation consists of:

- Reading all the required changes in the connected directory from the time of last retrieval
- Writing those changes into an import file. If the connected directory change records are not written in LDIF, then they are converted into LDAP change records.
- Keeping track of the last time the changes were extracted from the connected directory. This includes ensuring that the changes are retrieved before they are purged from the connected directory.
- Ensuring that only those changes in the connected directory required for synchronization with Oracle Internet Directory are written to the import file

Oracle Internet Directory Import Operation

An Oracle Internet Directory import operation consists of:

- Reading all the connected directory changes that the exporting agent wrote into the import file. If the import file is not written in LDIF, then this task includes converting the connected directory change records into LDAP change records.
- Updating the LDAP change records in Oracle Internet Directory.
- After all the changes are updated in Oracle Internet Directory, moving the import file to an archive directory

Synchronization Scenarios

Synchronization uses combinations of the import and export operations as described in the previous section.

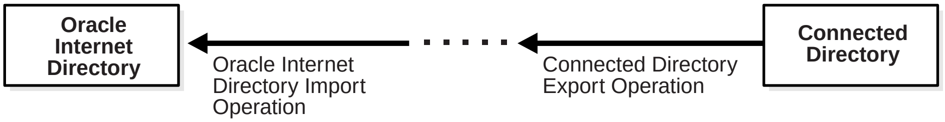
The exact operations involved in a given synchronization depend on whether changes are being applied from Oracle Internet Directory to a connected directory or the reverse.

Synchronizing from a Connected Directory to Oracle Internet Directory

This synchronization involves performing these operations in the following sequence:

1. Connected directory export operation
2. Oracle Internet Directory import operation

The following diagram illustrates the direction in which data flows in each operation, from a connected directory to Oracle Internet Directory.

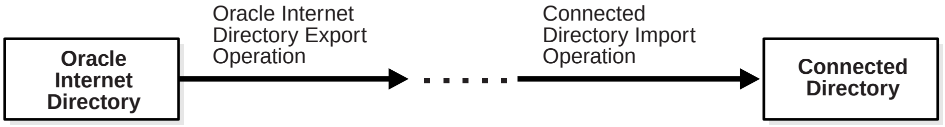

Oracle Internet Directory
Oracle Internet Directory Import Operation
Connected Directory Export Operation
Connected Directory

Synchronizing from Oracle Internet Directory to a Connected Directory

This synchronization involves performing these operations in the following sequence:

1. Oracle Internet Directory export operation
2. Connected directory import operation

The following diagram illustrates the direction in which data flows in each operation, from Oracle Internet Directory to a connected directory.


Oracle Internet Directory
Oracle Internet Directory Export Operation
Connected Directory Import Operation
Connected Directory

Although an agent can perform one or many of the four operations discussed earlier in this section, it typically performs only connected directory import and connected directory export operations. It relies on the directory integration server to perform the Oracle Internet Directory import and Oracle Internet Directory export operations.

To exchange data between itself and the directory integration server, an agent uses import and export files. If an agent is designed to perform a complete synchronization by using its own resources, then it can bypass these files.

The Oracle directory integration server can perform Oracle Internet Directory import and export operations, including attribute mappings. Agents do not need to perform these operations. In addition, the directory integration server can schedule the execution of agents.

Types of Agents

Depending on how it is deployed in the Oracle Directory Integration platform,an agent is known as either a **partner agent** or an **external agent**.

Partner Agents

Partner agents use the services of the directory integration server to perform the Oracle Internet Directory import and export operations. Moreover, the directory integration server controls their execution.

In a typical synchronization, a partner agent performs either the connected directory import operation or the connected directory export operation. The Oracle directory integration server performs the Oracle Internet Directory import and export operations. However, agents may also perform tasks that the directory integration server would otherwise do. For example, an agent may itself map attributes instead of relying on the directory integration server to do it.

Before you can use a partner agent with the Oracle Directory Integration platform**,** you must register it with Oracle Internet Directory. To do this, you create a **directory integration profile** in Oracle Internet Directory by using either Oracle Directory Manager or command-line tools.

Partner agents performing export operations do not need to worry about changes getting purged before they are consumed. Oracle Internet Directory maintains state information about changes applied by various agents and preserves that information until all partner agents have consumed the changes.

External Agents

Unlike partner agents, external agents are independent of the directory integration server when they perform Oracle Internet Directory export and import operations. Such agents are, for example, those that rely on third-party metadirectory engines for the same kinds of services that the directory integration server performs for partner agents.

Typically, an external agent performs a complete import or export synchronization. An external agent synchronizing from a connected directory to Oracle Internet Directory performs both the connected directory export and the Oracle Internet Directory import operations. Similarly, when synchronizing from Oracle Internet Directory to a connected directory, it performs both the Oracle Internet Directory export and the connected directory import operations.

External agents do not use the services of the directory integration server to synchronize between Oracle Internet Directory and connected directories. You do not need to register them with Oracle Internet Directory.

In export operations, external agents must use the standard LDAP change log interface to access change information from Oracle Internet Directory. It is the responsibility of the external agents to consume the changes in Oracle Internet Directory before those changes are purged.

Change Log Interfaces

To synchronize changes in Oracle Internet Directory with those in connected directories, the Oracle Directory Integration platform uses agents to retrieve changes in Oracle Internet Directory. Changes in Oracle Internet Directory are available in a container, called `Change Log Container`. Changes in the change log container are uniquely identified by a change log number.

There are two interfaces for retrieving changes from Oracle Internet Directory, one for partner agents and one for external agents.

For partner agents, Oracle Internet Directory and the directory integration server keep track of changes already applied by an agent and those still pending. This is done by maintaining status information for each agent indicating the point up to which it has exported changes to the connected directory. This attribute, called `orcllastappliedchangenumber`, is in the integration profile for the agent.

Oracle Internet Directory purges changes only after partner agents consume them.

In an export operation, the directory integration server updates the `orcllastappliedchangenumber` attribute for the agent only after it successfully runs the agent. The directory integration server performs data mappings, then writes changes from Oracle Internet Directory into the export file. Agents then consume the changes by reading the export file.

For external agents, the directory server does not maintain status information. For such agents, the attribute `orcllastchangenumber` in the **root directory specific entry** indicates the last change generated by the directory integration server.

Oracle Internet Directory makes changes available to external agents only for a period of time, after which it purges the changes. External agents must maintain their own status information about changes they have consumed and those still pending. They must consume the changes before the changes are purged.

To access changes in Oracle Internet Directory, external agents query the Oracle Internet Directory change log container. Typically, an external agent first retrieves the `orcllastchangenumber` attribute from the DSE root. Then, based on the value of `orcllastchangenumber` and the number of the last change applied, the external agent pulls changes not yet applied.

To find the last change number in Oracle Internet Directory, search the Oracle Internet Directory DSE root with a required attribute of `orcllastchangenumber`. Use these specifications for the search:

```
SCOPE : BASE
BASEDN : “”
FILTER: ‘(objectclass=*)’
REQUIRED ATTRIBUTE: orcllastchangenumber
```

To read a change log from Oracle Internet Directory, search with these specifications:

```
SCOPE : BASE
BASEDN : “cn=changelog”
FILTER:
‘(&(objectclass=changelogentry)(server=server-name)(changenumber>=change#))’
```

Registration of Partner Agents into Oracle Directory Integration Platform

Before deploying a partner agent, you register it in Oracle Internet Directory. This registration involves creating a directory integration profile in the directory. This integration profile is stored as an LDAP entry in the directory. To create it, you can use either Oracle Directory Manager or command-line tools.

Attributes in an integration profile entry belong to an object class called orclodiProfile. The only exception is the `orcllastChangeLogNumber` attribute, which belongs to the object class `orclChangeSubscriber`.

The Object ID prefix 2.16.840.1.113894.7 is assigned to platform-related classes and attributes. The following table lists all the attributes in the Oracle Directory Integration platform profile.

Table 23–1 Attributes in the Oracle Directory Integration Platform Profile

Attribute	Description
General Information	
Agent Name (orclODIPAgentName)	Name of the agent. This is used as an RDN component of the DN that identifies the integration profile. The name can contain only alpha-numeric characters.
Agent Control (orclODIPAgentControl)	Indicator of whether the agent is enabled or disabled. Valid values are ENABLE and DISABLE.
Agent Password (orclODIPAgentPassword)	Password that the directory integration server uses to bind to Oracle Internet Directory on behalf of the agent
Agent Host Name (orclODIPAgentHostName)	Host on which the agent runs
Synchronization Mode (orclODIPSynchronizationMode)	Direction of synchronization between Oracle Internet Directory and a connected directory. IMPORT indicates importing changes from the connected directory to Oracle Internet Directory. EXPORT indicates exporting changes from Oracle Internet Directory
Scheduling Interval (orclODIPSchedulingInterval)	Number of seconds after which a connected directory is synchronized with Oracle Internet Directory
Number of Retries (orclODIPSyncRetryCount)	Maximum number of retries that the directory integration server performs before disabling synchronization.
Execution Information	
Agent Execution Command (orclODIPAgentExeCommand)	Agent executable name and argument list used by the directory integration server
Connected Directory Account (orclODIPConDirAccessAccount)	Account used by the agent for accessing the connected directory. It is passed by the directory integration server to the agent specified at the command line when the agent is invoked.

Table 23–1 Attributes in the Oracle Directory Integration Platform Profile

Attribute	Description
Connected Directory Account Password (`orclODIPConDirAccessPassword`)	Password to be used by the agent when accessing the connected directory. It is passed by the directory integration server to the agent specified at the command line when the agent is invoked.
Agent Configuration Information (`orclODIPAgentConfigInfo`)	Any configuration information which an agent wishes to store in Oracle Internet Directory. It is passed by the directory integration server to the agent specified at the command line when the agent is invoked. This information is stored as a binary attribute. The directory integration server does not modify this attribute, but passes it directly to the specified agent.
Datafile Format (`orclODIPDatafileType`)	The type of the import or export file, either `LDIF` or `TAGGED`
Mapping Information	
Subscribed Domain (`orclODIPChangeSubscriptionDomain`)	DN of the subtree in Oracle Internet Directory to which an agent subscribes for all the changes it is to export
DN Construct Rule (`orclODIPEntryDNConstructRule`)	Rule for generating the DN of an entry in Oracle Internet Directory from its RDN during an import operation. For example, you could specify that, for entries of the form `cn=%s, dc=my_company, dc=com`, the `%s` is to be replaced by the actual RDN value.
Synchronization Key (`orclODIPSynchronizationKey`)	Attribute that uniquely identifies records in a connected directory. This is used as a key to synchronize Oracle Internet Directory and the connected directory.
Attribute Mapping Rules (`orclODIPAttributeMappingRules`)	Mapping rules for converting data from a connected directory to Oracle Internet Directory. This information is stored as a binary attribute. **See Also:** "Default Oracle Human Resources Agent Mapping Rules" on page 27-13 for an example of mapping rules

Table 23–1 Attributes in the Oracle Directory Integration Platform Profile

Attribute	Description
Mapping Filter (orclODIPMappingFilter)	Filter for excluding changes in Oracle Internet Directory that a connected directory does not require
Status Information	
Next Synchronization Time (orclODIPNextSynchronizationTime)	Time when the agent is to be executed next. Its format is dd-mon-yyyy hh:mm:ss, where *hh* is the time of day in a 24-hour format.
Synchronization Status (orclODIPSynchronizationStatus)	Execution status of the agent
Synchronization Errors (orclODIPSynchronizationErrors	Error message for the last error encountered. This is a multivalued attribute.
Con Dir Last Applied Change Time (orclodipConDirLastAppliedChgTime)	Time when the last change from the connected directory was applied to Oracle Internet Directory.Its format is dd-mon-yyyy hh24:mi:ss. The default is 01-Jan-2001 00:00:00 This attribute is mandatory. You can modify this attribute.
Con Dir Last Applied Change Num (orclodipConDirLastAppliedChgNum)	For agents performing import operations, indicates the last change from the connected directory that has been applied to Oracle Internet Directory.
OID Last Applied Change Number (orclOIDLastAppliedChgNum)	For export agents, the last change from Oracle Internet Directory that has been applied to the connected directory

The various integration profile entries in the directory are created under the container cn=subscriber profile, cn=changelog subscriber, cn=oracle internet directory. For example, an agent called OracleHRAgent is stored in the directory as orclodipagentname=OracleHRAgent, cn=subscriber profile,cn=changelog subscriber,cn=oracle internet directory.

Agent Configuration Information

An agent may need some configuration information at runtime for performing various operations. For example, to make it easier for users to specify which connected directory attributes are to be synchronized with Oracle Internet Directory, you may want an agent to store a list of these attributes as part of its configuration information. This kind of information is called agent configuration information.

You can store agent configuration information wherever and however you want. However, the Oracle Directory Integration platform enables you to store it as a binary attribute, called `orclODIPAgentConfigInfo`, in the integration profile. The Oracle directory integration server passes this information as a temporary file to the agent at the time of the agent’s invocation.

Agent configuration information is optional. If an agent does not require such information, then the corresponding attribute in the integration profile is left empty.

This configuration information can pertain to the agent or the connected directory or both. Oracle Internet Directory and the directory integration server do not read or modify this information, but pass it directly to the agent.

See Also:

- "File Naming Conventions" on page 23-16 for the names of these files
- "Location of Files" on page 23-16 for the location of these files

Mapping Rules

Mapping rules govern the conversion of attributes between a connected directory and Oracle Internet Directory. There is one set of mapping rules for each connected directory. This set is stored as a binary value in an attribute called `orclODIPAttributeMappingRules` in the integration profile in Oracle Internet Directory.

The directory integration server uses these rules to map attributes, as necessary, when generating an export file or interpreting an import file. When the directory integration server imports changes into Oracle Internet Directory, it converts the connected directory change records into LDAP change records, following the mapping rules specified in the integration profile. Similarly, when the directory integration server exports changes from Oracle Internet Directory, it converts the Oracle Internet Directory change records into connected directory change records, following the mapping rules specified in the integration profile.

An agent is not required to use the mapping function of the directory integration server. This could be the case, for example, when an agent does not use the import or export file interfaces, or when it does use import or export files of type LDIF. In such cases, the agent performs its own mappings and the `orclODIPAttributeMappingRules` attribute in the integration profile is left empty.

The Oracle Directory Integration platform supports both one-to-many and many-to-one mappings.

One-to-many mapping	One attribute in a connected directory can map to many attributes in Oracle Internet Directory. For example, suppose an attribute in the connected directory is `Address:123 Main Street/MyTown, MyState 12345`. You can map this attribute in Oracle Internet Directory to both the LDAP attribute `homeAddress` and the LDAP attribute `postalAddress`.
Many-to-one mapping	Multiple attributes in a connected directory may map to one attribute in Oracle Internet Directory. For example, suppose that the Human Resources directory represents Anne Smith by using two attributes: `firstname=Anne` and `lastname=Smith`. You can map these two attributes to one attribute in Oracle Internet Directory: `cn=Anne Smith`.

Mapping Rules Format

Mapping rules are organized in a fixed tabular format, and you must follow that format carefully. The fields are delimited by a colon (:). The first line consists of fixed column headers. Do not change the column names. For each `conndirattrname` and `oidattrname` pair, you define only one mapping.

Each record in the mapping configuration file uses the following format:

```
OIDCLASSNAME:OIDATTRNAME:OIDATTRTYPE:CONNDIRCLASSNAME:CONNDIRATTRNAME:CONNDIRATT
RTYPE:MAPPINGRULE
```

Table 23–2 describes the columns.

Table 23–2 Columns in the Mapping Configuration File

Column Name	Description
OIDCLASSNAME	Object class of the Oracle Internet Directory attributes
OIDATTRNAME	Attribute name of the attribute in Oracle Internet Directory
OIDATTRTYPE	Attribute type of the Oracle Internet Directory attribute
CONNDIRCLASSNAME	Object class of the connected directory attributes
CONNDIRATTRNAME	Attribute name of the connected directory attribute
CONNDIRATTRTYPE	Attribute type of the connected directory attributes
MAPPINGRULE	Mapping rule to use when importing or exporting data

The following table lists and describes the mapping rules for importing into Oracle Internet Directory:

Mapping Rule	Description
`COPY_STRING`	Copy the source attribute value string (as indicated by `localAppAttrName`) to the destination attribute value (as indicated by `ldapAttrName`).
`COPY_STRING,`*`arg1`*	Copy the source attribute value string indicated by the argument *`arg1`* after the comma (,) to the destination attribute indicated by the `ldapAttrName` value.
`COPY_STRING_LOWER`	Copy the source attribute from `localAppAttrName` to the destination attribute `ldapAttrName` value after converting it to lowercase.
`COPY_STRING_LOWER,`*`arg1`*	Copy the source attribute from *`arg1`* value string to the destination attribute `ldapAttrName` after converting it to lowercase.
`COPY_STRING_UPPER`	Copy the source attribute `localAppAttrName` to the destination attribute `ldapAttrName` value after converting it to uppercase.
`COPY_STRING_UPPER,`*`arg1`*	Copy the source attribute *`arg1`* to the destination attribute `ldapAttrName` after converting it to uppercase.

Mapping Rule	Description
`APPEND_STRING,`*`arg1`*`,`*`arg2`*	Append the value of source attribute *`arg1`* to the already existing destination attribute value by using concatenation separator *`arg2`*.
`TRIM_STRING,`*`arg1`*`,`*`arg2`*	Copy the value generated by truncating the value of source attribute *`arg1`* at character *`arg2`* to the destination attribute `ldapAttrName`.
`LITERAL`	Copy the literal value indicated by the argument after the comma (,) value string to the destination attribute `ldapAttrName` value.

> **See Also:** "Default Oracle Human Resources Agent Mapping Rules" on page 27-13 for an example of mapping rules

Import and Export Files

These files store data extracted from either a connected directory or Oracle Internet Directory. The platform uses them to exchange data between Oracle Internet Directory and connected directories.

Import files contain changes from the connected directory. Export files contain changes from Oracle Internet Directory.

Oracle Internet Directory release 3.0.1 supports tagged and LDIF files only.

Tagged Files In these files, each record consists of a tag and value pair separated by a colon (:). A multivalued attribute is represented by multiple rows with the same tag.

The following example of a tagged file contains attributes of an employee record:

```
FirstName:John
LastName:Liu
EmployeeNumber:12345
Title:Mr.
Sex:M
MaritalStatus:Married
TelephoneNumber:123-456-7891
Mail:Jliu@my_company.com
Address:100 Jones Parkway
City:MyTown
```

LDIF Files A partner agent can exchange data with the directory integration server by using an LDIF file. In this case, the agent—not the directory integration server—performs the attribute mappings.

In an import operation from a connected directory into Oracle Internet Directory, the agent can map attributes and generate the import file in LDIF for the directory integration server. In an export operation from Oracle Internet Directory into a connected directory, the directory integration server can create an export file in LDIF, leaving the agent to map the attributes.

File Naming Conventions

All filenames correspond to the name of the agent, as in the following table:

File	Filename
Data file	`Agent_Name.data`
Error file	`Agent_Name.err`
Agent configuration file	`Agent_Name.conf`
Mapping rules file	`Agent_Name.map`

For example, the datafile name of the Oracle Human Resources agent is `oraclehragent.data`.

Location of Files

This table tells you where to find the various files:

Files	Path Name
Import files	`$ORACLE_HOME/ldap/odi/data/import`
Export files	`$ORACLE_HOME/ldap/odi/data/export`
Error files	`$ORACLE_HOME/ldap/odi/log`
Configuration and mapping files	`$ORACLE_HOME/ldap/odi/conf`

Managing Partner Agents

This section contains these topics:

- Managing Partner Agents by Using Oracle Directory Manager
- Managing Partner Agents from the Command Line

Managing Partner Agents by Using Oracle Directory Manager

This section tells you how to register and deregister a partner agent by using Oracle Directory Manager.

Registering a Partner Agent by Using Oracle Directory Manager

Oracle Directory Manager enables you to register a partner agent in one of two ways:

- By creating a new configuration set entry, then adding an integration profile to it
- By selecting an existing configuration set entry, then adding an integration profile to it

To register an agent:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Server Management, then select Directory Integration Server. The Active Processes box appears in the right pane.
2. On the toolbar, click Create. The Configuration Sets dialog box appears.
3. In the Configuration Sets dialog box, click Create. The Integration Profiles dialog box appears. You have two options:
   - To create an integration profile by copying an existing one, select the Oracle Directory Integration platform profile you want to copy, then click Create Like. The Integration Profile dialog box displays the General tab page.
   - To create an integration profile without copying an existing one, click Create New. The Integration Profile dialog box displays the General tab page.

4. In the General tab page, fill in the fields as explained in Table 23–3.

Table 23–3 Description of Fields on the General Tab Page in Oracle Directory Manager

Field	Description
Agent Name	Specify the name of the agent. The name you enter is used as the RDN component of the DN for this integration profile. For example, specifying an agent name `MSAccess` creates an integration profile named `orclmetaconnname=MSAccess, cn=subscriber profile, cn=changelog subscriber,cn=oracle internet directory`. This field is mandatory. There is no default.
Synchronization Mode	Specify whether this is an import or an export operation. An import operation pulls changes from a connected directory into Oracle Internet Directory. An export operation pushes changes from Oracle Internet Directory into a connected directory. This field is mandatory. The default is `IMPORT`. **Note:** Oracle Internet Directory release 3.0.1 supports the import synchronization mode only.
Agent Control	Specify whether the agent is enabled or disabled. This field is mandatory. The default is `ENABLED`.
Agent Password	Specify the password that the directory integration server is to use when binding to Oracle Internet Directory on behalf of the agent. This field is mandatory. The default is `welcome`.
Host Name	Specify the host on which the agent will run. This field is mandatory and there is no default.
Number of Retries	Specify the maximum number of times the directory integration server is to attempt synchronization before it disables synchronization. This field is mandatory. The default is 5.
Scheduling Interval	Specify the number of seconds between synchronization attempts between a connected directory and Oracle Internet Directory. This field is mandatory. The default is `60`.

5. Select the Execution tab and fill in the fields as explained inTable 23–4.

Table 23–4 Description of Fields on the Execution Tab in Oracle Directory Manager

Field	Description
Execution Command	Specify the agent executable name and the arguments used by the directory integration server to execute the agent. This field is mandatory. There is no default.
Connected Directory Account	Specify the account to be used by the agent for accessing the connected directory. For example, if the connected directory is a database, the account might be `Scott`. If the connected directory is another LDAP-compliant directory, then the account might be `cn=Directory Manager`. This field is optional. There is no default.
Connected Directory Account Password	Specify the password the agent is to use when accessing the connected directory. This field is optional. There is no default.
Agent Config Info	This field displays additional information that the directory integration server passes to an agent. You cannot modify this field. There is no default.
Datafile Format	The format used by the import or export file. Valid values are `LDIF` or `TAGGED`. This field is optional. The default is `TAGGED`.

6. Select the Mapping tab and fill in the fields as explained in Table 23–5.

Table 23–5 Description of Fields on the Mapping Tab in Oracle Directory Manager

Field	Description
Attribute Mapping Rules	This field displays the mapping rules for converting data between a connected directory and Oracle Internet Directory. There is no default. **Note:** You cannot edit the mapping rules file by using Oracle Directory Manager. You edit the mapping rules file manually and then upload it to the profile by using the provided script, `ldapCreateConn.sh`.
Synchronization Key	Specify the attribute that uniquely identifies records in a connected directory. This is used as a key to synchronize Oracle Internet Directory and the connected directory. This field is optional.
Subscribed Domain	Specify the DN of the Oracle Internet Directory subtree from which an agent is to export changes. This field is optional. There is no default.

Table 23–5 Description of Fields on the Mapping Tab in Oracle Directory Manager

Field	Description
DN Construct Rule	Specify the rule for generating the DN from the RDN of an Oracle Internet Directory entry. For example, in the rule `cn=%s, dc=acme,dc=com`, the `%s` is replaced by an actual RDN value. This field is optional. There is no default.
Mapping Filter	Specify a filter for excluding changes in Oracle Internet Directory that a connected directory does not require. There is no default.

7. Select the Status tab and fill in the fields as explained in Table 23–6.

Table 23–6 Description of Fields on the Status Tab in Oracle Directory Manager

Field	Description
OID Last Applied Change Number	For export operations, specify the identifier of the last change from Oracle Internet Directory that has been applied to the connected directory. The default is `0`.
Next Synchronization Time	The next absolute time that the agent is to be executed. The default is the time at which the agent is created.
Synchronization Status	The execution status of the agent. You cannot modify this field. The default is `YET TO BE EXECUTED`.
Synchronization Errors	The last error message. You cannot modify this field. There is no default.
Last Applied Change Number	Pertains to import operations. This field displays the number of the last change applied from a connected directory to Oracle Internet Directory. You cannot modify this field. The default is `0`.
Last Synchronization Time	Pertains to export operations. This field displays the time when the last change from Oracle Internet Directory was applied to the connected directory. The default is the time at which the agent is created.

8. In the Integration Profile dialog box, click OK. This returns you to the Configuration Sets dialog box, which now lists the integration profile you just created.

9. Click OK to exit the Configuration Sets dialog box. The agent you created is now registered with Oracle Internet Directory.

Deregistering a Partner Agent by Using Oracle Directory Manager

To delete an agent:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*> Server Management > Directory Integration Server.
2. Select the Configuration Set from which to delete the agent. The Integration Profiles tab page appears in the right pane.
3. In the Integration Profiles tab page, select the agent you want to deregister, then click Delete.

Managing Partner Agents from the Command Line

This section tells you how to register and deregister agents by using the script `ldapcreateConn.sh`.

Registering a Partner Agent by Using ldapcreateConn.sh

You can register an agent by using the command-line tool `ldapcreateConn.sh`. This tool is in the directory `$`*`ORACLE_HOME/`*`ldap/admin/`.

The following example registers an agent named `HRMS` in configuration set 2 (`config 2`):

```
ldapcreateConn.sh name HRMS [ -host MyHost] [port 389] binddn cn=orcladmin pass
welcome data TST -acct apps -pwd apps -ldapctx dc=hr,dc=metadirectory,dc=com
config 2
```

Table 23–7   Arguments for Registering a Partner Agent by Using ldapcreateConn.sh

Argument	Description
`name`	Name of the agent. This is used as the RDN of the integration profile entry
`-host`	Host name of the directory server
`-port`	Port number on which the directory server is running. Default is 389.
`binddn`	The bind DN with which the tool binds to the directory. The bind DN must have the privilege to add integration profile entries.
`pass`	Password of the entry referred by the bind DN
`-acct`	Account name in the connected directory that will be used by the agent to connect to the connected directory

Table 23–7 Arguments for Registering a Partner Agent by Using ldapcreateConn.sh

Argument	Description
`-pwd`	Password of the connected directory account
`-ldapctx`	The parent DN where the integration profile entry is created
`config 2`	The configuration set entry of the directory integration server with which this integration file is associated

When the integration server is invoked for configuration set 2, this agent is run. You can see a full description by invoking ldapCreateConn.sh with the `-help` argument.

Deregistering a Partner Agent Using ldapdeleteConn.sh

You can deregister a agent by using the command-line tool `ldapdeleteConn.sh`. This tool is in the directory `$`*`ORACLE_HOME`*`/ldap/admin/`.

The following example deregisters an agent entry and dissociates it from the configuration set 2 (`config 2`) entry:

```
ldapdeleteconn.sh name HRMS config 2
```

24

Managing the Oracle Directory Integration Server

This chapter discusses the Oracle directory integration server and tells you how to configure and manage it. It contains these topics:

About the Oracle Directory Integration Server

The Oracle directory integration server is the central component of the Oracle Directory Integration platform. It is a daemon server process that does the following:

Schedules agents

The directory integration server controls the execution of agents, invoking them at specified times. The scheduling information is stored in the integration profile associated with the agent. When it invokes agents, the directory integration server also passes agent configuration information to them.

Imports and exports data

The directory integration server imports and exports changes into and out of Oracle Internet Directory. The directory integration server and the agents exchange change information by using import files and export files.

Maps attributes

The directory integration server includes a generic mapping facility for performing attribute mappings. It maps attributes based on a set of rules that you specify in Oracle Internet Directory. The directory integration server maps attributes either when generating an export file during an export operation, or when interpreting an import file during an import operation.

You can run multiple directory integration server instances.

Only partner agents use the directory integration server. External agents do not use it.

This section contains these topics:

- The Oracle Directory Integration Server and Configuration Set Entries
- Configuration Data Refresh
- LDAP Connections Used by the Oracle Directory Integration Server
- Registering the Oracle Directory Integration Server

The Oracle Directory Integration Server and Configuration Set Entries

When you start the directory integration server by using the **OID Control Utility**, the start message you send refers to a **configuration set entry** containing server parameters. That configuration set is, in turn, associated with one or many agents. The directory integration server runs the agents associated with the particular configuration set.

The server has four types of threads of execution in the process:

Controller thread	Monitors all the other threads
Configuration reader thread	Periodically polls for changes in the directory integration profiles in the directory, then refreshes the directory integration profiles in its cache with that information. The default polling interval, in minutes, is 2.
Agent threads	Spawn the agent executable and mapping service as subprograms. These threads are created only for executing the agents. They terminate when the synchronization cycle for the agent is over.
Scheduler thread	Schedules the agents for execution. Every time a timer is triggered, it spawns an agent thread.

If there are no agents configured for the configuration set, or if all the configured agents are disabled, then the Oracle Directory Integration server does not initiate synchronization. Instead, it waits indefinitely for agents to be added to that configuration set. If the configuration set specified at the command line does not exist in the directory, then the Oracle Directory Integration server logs this information in the log file and exits.

Configuration Data Refresh

Agent configuration data is checked every 2 minutes for changes by the configuration reader thread, and the entire configuration data cache is refreshed in memory as required. The server, if started with the proper debug level, can give the appropriate messages.

See Also:

- "Managing Server Configuration Set Entries" on page 6-2 for more information on configuration set entries
- "Managing Partner Agents by Using Oracle Directory Manager" on page 23-17 for instructions on enabling and disabling directory integration agents
- "Setting the Debug Level" on page 24-14 for more information about debug levels

LDAP Connections Used by the Oracle Directory Integration Server

Whenever it executes an agent at synchronization time, the directory integration server starts an agent thread. This thread opens an LDAP connection to the directory server, then closes the connection before exiting.

In addition, the configuration reader thread uses one LDAP connection for periodically refreshing its cache with configuration information from Oracle Internet Directory.

Registering the Oracle Directory Integration Server

After installing the directory integration server, you must register it with Oracle Internet Directory. You must separately register each directory integration server installed on a different host. You do this by using the Oracle directory integration server registration tool (`odisrvreg`).

To run this tool, you need the privileges of an Oracle Internet Directory administrator. Run the tool from the machine on which the directory integration server is installed.

The tool creates an entry in the directory as part of the registration. It sets the password for the directory integration server and stores it as an encrypted value in the registration entry. If the registration entry already exists, then you can use the tool to reset the existing password. You must supply the correct password to run the tool.

In addition to generating the registration entry in the directory, the tool also creates a local file, called `odisrvwallet`, that acts as a private wallet for the directory integration server. The directory integration server, when it starts, uses this file to bind to the directory. It creates this file in the `$`*`ORACLE_HOME`*`/ldap/odi/conf` directory.

You can run the tool in SSL mode to make communication between the tool and the directory fully secure.

To register the directory integration server, enter this command:

```
odisrvreg -h hostname -p port -D binddn  -w bindpasswd
```

Table 24–1 Descriptions of ODISRVREG arguments

Argument	Description
-h hostname	Oracle directory server host name
-p *port_number*	Port number on which the directory server is running
-W *binddn*	Bind DN. The bind DN must have authorization to create the registration entry for the directory integration server.
-W *bindpasswd*	Bind password

To run the Oracle directory integration server registration tool in the SSL mode, enter the following:

```
odisrvreg -h hostname -p port -D binddn  -w bindpasswd -U ssl_mode -W wallet -P
wallet_password
```

Table 24–2 Descriptions of ODISRVREG arguments

Argument	Description
-h hostname	Oracle directory server host name
-p *port_number*	Port number on which the directory server is running
-W *binddn*	Bind DN. The bind DN must have authorization to create the registration entry for the directory integration server.
-W *bindpasswd*	Bind password
`-U` *`ssl mode`*	SSL mode. For no authorization, specify `0`. For one-way authorization, specify `1`.

Table 24–2 Descriptions of ODISRVREG arguments

Argument	Description

`-W wallet`	SSL wallet. Enter the full path. For example, on Solaris, you could set this parameter as follows:
	`file:/home/my_dir/my_wallet`
	On Windows NT, you could set this parameter as follows:
	`file:C:\my_dir\my_wallet`

`-P wallet password`	Password for opening the SSL wallet

Managing Configuration Set Entries

When it starts, the directory integration server needs a list of all the agents that the directory integration server is to control. A configuration set entry holds this information for the directory integration server. You can create, modify, and view configuration set entries by using either Oracle Directory Manager or the appropriate command line tools.

When an agent is registered, an integration profile is created in the directory for that agent. The integration profile is always associated with a configuration set entry. In this way, the association between an agent and the Directory Integration Server is established.

When you start the directory integration server, a configuration set entry is supplied as part of the argument list. This configuration set entry determines the behavior of the directory integration server.

You can control the runtime behavior of the directory integration server by using a different configuration set entry when you start it. For example, you can start instance 1 of the directory integration server on host H1 with `configset1`, and instance 2 of the directory integration server on host H1 with `configset2`. The behavior of instance 1 of the directory integration server depends on configset 1, and that of instance 2 depends on configset2. By dividing different agents on host H1 between the two configuration set entries, you are distributing the load of running the agents on host H1 between the two directory integration server instances.

Managing the Oracle Directory Integration Server

This section contains these topics:

- Starting the Oracle Directory Integration Server
- Stopping the Oracle Directory Integration Server
- Using the Restart Command
- Using the Oracle Directory Integration Server in SSL Mode
- Finding the Log File
- Setting the Debug Level
- Changing the Synchronization Status Attribute

Starting the Oracle Directory Integration Server

The Oracle directory integration server executable, `odisrv`, resides in the `$`*`ORACLE_HOME`*`/bin` directory.

The way you start the directory integration server depends on whether your installation includes the **OID Monitor** and the **OID Control Utility**. These tools—along with other server and client components—are parts of a typical installation. In such installations, you start the directory integration server by using these tools.

> **Note:** Although you can start the directory integration server without using the OID Monitor and the OID Control Utility, Oracle Corporation recommends that you use them. This way, if the directory integration server unexpectedly terminates, then the OID Monitor automatically restarts it.

Client-only installations do not include the OID Monitor and the OID Control Utility. In such installations, you start the directory integration server from the command line.

Starting the Oracle Directory Integration Server by Using OID Monitor and the OID Control Utility

To start the directory integration server:

1. Be sure that OID Monitor is running. To verify this, enter the following at the command line:

   ```
   ps -ef | grp oidmon
   ```

   If OID Monitor is not running, then start it by following the instructions in "Task 1: Start the OID Monitor" on page 4-2.

2. Start the directory integration server by using the OID Control utility by entering:

   ```
   oidctl [connect=net_service_name] server=odisrv [instance=instance_number]
   config=configuration_set_number [flags="[host=hostname] [port=port_number]
   [debug=debug_level]"] start
   ```

   The arguments in this command are described in the following table.

Table 24–3 Description of Arguments for Starting Oracle Directory Integration Server

Argument	Description
`connect=`*`net_service_name`*	If you already have a `tnsnames.ora` file configured, then this is the net service name specified in that file, located in $*ORACLE_HOME*/`network/admin`
`server=odisrv`	Type of server to start. In this case, the server you are starting is `odisrv`. This is not case-sensitive. This argument is mandatory.
`instance=`*`instance_number`*	Specifies the instance number to assign to the directory integration server. This instance number must be unique. OID Monitor verifies that the instance number is not already associated with a currently running instance of this server. If it is associated with a currently running instance, then OID Monitor returns an error message.
`config=`*`configuration_set_number`*	Specifies the number of the configuration set that the the directory integration server is to execute. This argument is mandatory.
`host=`*`hostname`*	Oracle directory server host name

Table 24–3 Description of Arguments for Starting Oracle Directory Integration Server

Argument	Description
`port=`*port_number*	Oracle directory server port number
`debug=`*debug_level*	The required debugging level of the directory integration server **See Also:** Table 24–6 on page 24-14 for a description of the various debug levels

Starting the Oracle Directory Integration Server Without Using OID Monitor and the OID Control Utility

To start the directory integration server, enter the following at the command line:

```
odisrv [host=host_name] [port=port_number]
config=configuration_set_number [instance=instance_number] [debug=debug_level]
```

Stopping the Oracle Directory Integration Server

Stop the directory integration server in one of two ways, depending on how you started it.

Stopping the Oracle Directory Integration Server by Using OID Monitor and the OID Control Utility

If you started the directory integration server by using OID Monitor and the OID Control utility, then you must stop it by using them.

To stop the directory integration server by using the OID Monitor:

1. Before you stop the directory integration server, be sure that the OID Monitor is running. To verify this, enter the following at the command line:

   ```
   ps -ef | grp oidmon
   ```

   If OID Monitor is not running, then start it by following the instructions in "Task 1: Start the OID Monitor" on page 4-2.

2. Stop the directory integration server by entering:

   ```
   oidctl [connect=net_service_name] server=odisrv instance=instance stop
   ```

Stopping the Directory Integration Server Without Using OID Monitor and the OID Control Utility

If you started the directory integration server without using OID Monitor and the OID Control utility, then you must stop it by using the command line.

To stop the directory integration server by using the command line:

1. Enter the following at the command line to determine the PID (process identifier) of the directory integration server:

   ```
   ps –ef | grep odisrv
   ```

2. Stop the directory integration server by entering the following at the command line:

   ```
   kill PID
   ```

Using the Restart Command

If you use OID Monitor and the OID Control utility, then you can both stop and restart the directory integration server in one command, namely, `restart`. This is useful when you want to refresh the server cache immediately, rather than at the next scheduled time. When the directory integration server restarts, it maintains the same parameters it had before it stopped.

To restart the directory integration server:

1. Make sure that OID Monitor is running. To verify this, enter the following at the command line:

   ```
   ps -ef | grp oidmon
   ```

   If OID Monitor is not running, then start it by following the instructions in "Task 1: Start the OID Monitor" on page 4-2.

2. At the command line, enter:

   ```
   oidctl [connect=net_service_name] server=odisrv instance=instance_number
   restart
   ```

Using the Oracle Directory Integration Server in SSL Mode

To secure the data exchanged between Oracle Internet Directory and the directory integration server, you run both the directory server and the directory integration server in SSL mode.

Starting the Oracle Directory Integration Server in SSL Mode by Using OID Monitor and OID Control

To run the directory integration server in the SSL mode by using OID Monitor and the OID Control utility, enter the following command:

```
oidctl [connect=net_service_name] server=odisrv [instance=instance_number]
config=configuration_set_number [flags= [host=hostname] [port=port_number]
[debug=debug_level] [sslauth=<ssl mode> wloc= <wallet> wpass=<wallet
password>"]]start
```

Table 24–4 describes the arguments in this command.

Table 24–4 Description of Arguments for Starting Oracle Directory Integration Server in SSL Mode by Using OID Monitor and OID Control

Argument	Description
connect=*net_service_name*	If you already have a tnsnames.ora file configured, then this is the net service name specified in that file, located in $*ORACLE_HOME*/network/admin
server=odisrv	Type of server to start. In this case, the server you are starting is odisrv. This is not case-sensitive. This argument is mandatory.
instance=*instance_number*	Specifies the instance number to assign to the directory integration server. This instance number must be unique. OID Monitor verifies that the instance number is not already associated with a currently running instance of this server. If it is associated with a currently running instance, then OID Monitor returns an error message.
config=*configuration_set_number*	Specifies the number of the configuration set that the the directory integration server is to execute. This argument is mandatory.
host=*hostname*	Oracle directory server host name
port=*port_number*	Oracle directory server port number
debug=*debug_level*	The required debugging level of the directory integration server **See Also:** Table 24–6 on page 24-14 for a description of the various debug levels
sslauth *ssl_mode*	SSL modes (0: NO Auth, 1: One Way)

Table 24–4 Description of Arguments for Starting Oracle Directory Integration Server in SSL Mode by Using OID Monitor and OID Control

Argument	Description

wloc *wallet*	SSL wallet. Enter the full path. For example, on Solaris, you could set this parameter as follows:
	`file:/home/my_dir/my_wallet`
	On Windows NT, you could set this parameter as follows:
	`file:C:\my_dir\my_wallet`

wpass *wallet_password*	Password used for opening the SSL wallet

Starting the Oracle Directory Integration Server in SSL Mode Without Using OID Monitor and OID Control

To start the directory integration server in SSL Mode without using OID Monitor and OID Control, enter this command:

```
odisrv [host=host_name] [port=port_number] config=configuration_set_number
[instance=instance_number] [debug=debug_level]  [sslauth=ssl_mode wloc=wallet
wpass=wallet_password]
```

Table 24–5 Description of Arguments for Starting Oracle Directory Integration Server in SSL Mode Without Using OID Monitor and OID Control

Argument	Description
`instance=`*`instance_number`*	Specifies the instance number to assign to the directory integration server. This instance number must be unique. OID Monitor verifies that the instance number is not already associated with a currently running instance of this server. If it is associated with a currently running instance, then OID Monitor returns an error message.
`config=`*`configuration_set_number`*	Specifies the number of the configuration set that the the directory integration server is to execute. This argument is mandatory.
`host=`*`hostname`*	Oracle directory server host name
`port=`*`port_number`*	Oracle directory server port number

Table 24–5 Description of Arguments for Starting Oracle Directory Integration Server in SSL Mode Without Using OID Monitor and OID Control

Argument	Description
`debug=`*`debug_level`*	The required debugging level of the directory integration server **See Also:** Table 24–6 on page 24-14 for a description of the various debug levels
sslauth *ssl_mode*	SSL modes (0: NO Auth, 1: One Way)

wloc *wallet*	SSL wallet. Enter the full path. For example, on Solaris, you could set this parameter as follows:
	`file:/home/my_dir/my_wallet`
	On Windows NT, you could set this parameter as follows:
	`file:C:\my_dir\my_wallet`

wpass *wallet_password*	Password used for opening the SSL wallet

> **Note:** Although you can start the directory integration server without using the OID Monitor and the OID Control Utility, Oracle Corporation recommends that you use them. This way, if the directory integration server unexpectedly terminates, then the OID Monitor automatically restarts it.

Finding the Log File

The log file is located in the `$`*`ORACLE_HOME/`*`ldap/log/Oidsync`*`Server_instance_number`*`.log` directory.

For example, if the server was started as server instance number 3, then the log file would have this path name: `$`*`ORACLE_HOME/`*`ldap/log/oidsync03.log`.

Setting the Debug Level

You can specify the kinds of events listed in a log file by using the `debug` flag.

To specify multiple types of debugging:

1. Add the numeric values of the individual types as indicated in Table 24–6 on page 24-14.
2. At the command line, specify the total value. For example, the following command sets the debug level to `484`:

   ```
   oidctl server=odisrv flags=”debug=484" start
   ```

The various types of debug types are listed in Table 24–6.

Table 24–6 Debug Types

Debug Event Type	Numeric Value
Starting and stopping of different threads. Process related.	4
Detail level. Shows the spawned commands and the command-line arguments passed	32
Operations being performed by configuration reader thread. Configuration refresh events.	64
Actual configuration reading operations	128
Operations being performed by scheduler thread in response to configuration refresh events, and so on	256
Creation of callout lists for timers for different agents	512
Spawned agent and command names	1024
Monitoring of spawned agent processes	2048
Debugging of mapping service built into the Oracle Directory Integration server	4096
Debugging of the agent executable	8192
Detail agent level tracing	32768
Debugging of LDAP operations	65536
Detailed debugging of mapping service built into the Oracle Directory Integration server	131072

If you do not set a value for the debug flag, then the default level is `0` (zero).

Each trace statement in the log file includes:

- Timestamp
- Thread type
- Agent name

The various trace-statement types are:

`OME:CTL`	Messages from the controller thread
`OME:CFG`	Messages from the configuration reader thread
`OME:SCH`	Messages from the scheduler thread
`OME:CONN`	Messages from the thread which executes the agent and the mapping service

Changing the Synchronization Status Attribute

In an export operation, the server constantly updates the synchronization status attribute, `orcllastappliedchangenumber`, while synchronization is in progress. In Oracle Directory Manager, this field is called *OID last applied change number*.

To change this attribute manually from Oracle Directory Manager:

1. Disable the agent by using Oracle Directory Manager.
2. Make the attribute changes.
3. Re-enable the agent after the change.

Viewing Oracle Directory Integration Server Information

When the directory integration server starts, it generates specific runtime information and stores it in the directory. This information includes:

- Instance number of the directory integration server
- Host on which it is running
- Configuration set with which the directory integration server was started
- State of the configuration set refresh flag. This flag indicates to the directory integration server whenever there is a change to a directory integration profile and a refresh is required.

You can view this information for the directory integration server by using either Oracle Directory Manager or ldapsearch.

The entry containing the runtime information for the directory integration server uses the following format:

```
cn=instance_number,cn=odisrv,cn=subregistrysubentry
```

This section contains these topics:

- Viewing Oracle Directory Integration Server Runtime Information by Using Oracle Directory Manager
- Viewing Oracle Directory Integration Server Runtime Information by Using ldapsearch

Viewing Oracle Directory Integration Server Runtime Information by Using Oracle Directory Manager

To view runtime information for the directory integration server instance by using Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Server Management, then select Directory Integration Server. The Active Processes box appears in the right pane.
2. Click View Properties. The Server Process dialog box displays the information.

Viewing Oracle Directory Integration Server Runtime Information by Using ldapsearch

To view registration information for the directory integration server instance by using ldapsearch, perform a base search on its entry. For example:

```
ldapsearch -p 389 -h my_host -b cn=instance1,cn=odisrv,cn=subregistrysubentry -s
base -v "objectclass=*"
```

This example search returns the following:

```
dn: cn=instance1,cn=odisrv,cn=subregistrysubentry
cn: instance1
orcldiaconfigdns: "orclDIAName=HR,cn=subscriber profile,cn=changelog subscriber,
cn=oracle internet directory"
orcldiaconfigrefreshflag: 0
orclhostname: my_host
orclconfigsetnumber: 1
objectclass: top
objectclass: orclDIA
```

25

Managing Security in the Oracle Directory Integration Platform

This chapter discusses the most important aspects of security in the Oracle Directory Integration platform. It contains these sections:

Authentication

Authentication is the process by which the Oracle directory server establishes the true identity of the user connecting to the directory. It occurs when an LDAP session is established by means of the ldapbind operation.

It is important that each component in the Oracle Directory Integration platform be properly authenticated before it is allowed access to the directory.

Secure Sockets Layer (SSL) and the Oracle Directory Integration Platform

You can deploy the Oracle Directory Integration platform either with or without **Secure Socket Layer (SSL)**. SSL implementation supports these modes:

- No authentication—Provides SSL encryption of data, but does not use SSL for authentication
- SSL server authentication—Includes both SSL encryption of data and SSL authentication of the server to the client. In the Oracle Directory Integration platform, the server is the directory server, the client is the directory integration server.

  The server verifies its identity to the client by sending a **certificate** issued by a trusted **certificate authority (CA)**. This mode requires a public key infrastructure (PKI) and SSL wallets to hold the certificates.

To use SSL with the Oracle Directory Integration platform, you must start both the Oracle directory server and the Oracle directory integration server in the SSL mode.

> **See Also:**
>
> - "Using the Oracle Directory Integration Server in SSL Mode" on page 24-10 for instructions on starting the directory integration server in SSL mode
> - Chapter 4, "Preliminary Tasks" for instructions on starting the Oracle directory server in SSL mode

Oracle Directory Integration Server Authentication

You can install and run multiple instances of the directory integration server on various hosts. When you do this, beware of a malicious user either posing as the directory integration server or using an unauthorized copy of it.

To avoid such security issues:

- Ensure that each directory integration server is identified properly
- Ensure that, when you start a directory integration server, it is properly authenticated before it obtains access to Oracle Internet Directory

Non-SSL Authentication

To use non-SSL authentication, register each directory integration server by using the registration tool called odisrvreg.

The registration tool creates:

- An identity entry in the directory. The directory integration server uses this entry when it binds to the directory
- An encrypted password. It stores this password in the directory integration server entry.
- A private wallet on the local host. This wallet contains the security credentials, including an encrypted password. The name of the wallet is `odisrvwallet`, and it is stored in the `$`*`ORACLE_HOME`*`/ldap/odi/conf` directory.

When it binds to the directory, the directory integration server uses the encrypted password in the private wallet.

Note: Ensure that the wallet is protected against unauthorized access.

See Also: "Registering the Oracle Directory Integration Server" on page 24-4 for instructions about registering the directory integration server

Authentication in SSL Mode

The identity of the directory server can be established by starting both Oracle Internet Directory and the directory integration server in the SSL server authentication mode. The directory server provides its certificate to the directory integration server, which acts the client of Oracle Internet Directory.

The directory integration server is authenticated by using the same mechanism used in the non-SSL mode.

Agent Authentication

Within Oracle Internet Directory, an agent is a user with its own DN and password. This information is stored in the integration profile of the agent. To protect the profile from unauthorized access, establish appropriate access control policies for it in the directory. Only the Oracle Directory Integration platform administrator or a user designated by the Oracle Internet Directory administrator can create the integration profiles.

When the directory integration server performs a task on behalf of an agent, it binds to the directory as that agent and uses the agent name and password stored in the agent profile. The Oracle Directory Integration platform uses this mechanism to authenticate agents in both the SSL and non-SSL mode.

Access Control and Authorization

Authorization is the process of ensuring that a user reads or updates only the information for which that user has privileges. When directory operations are attempted within a directory session, the directory server ensures that the user—identified by the authorization identifier associated with the session—has the requisite permissions to perform those operations. Otherwise, the operation is disallowed. Through this mechanism, the directory server protects directory data from unauthorized operations by directory users. This mechanism is called access control. Access control information is the directory metadata that captures the administrative policies relating to access control.

Access to data in Oracle Internet Directory is restricted for both the directory integration server as well as the agents only to the desired subset of data by placing appropriate access policies in the directory. The following section discusses these policies in detail.

Access Controls for the Oracle Directory Integration Server

The directory integration server binds to the directory both as itself and on behalf of the agent.

- When it binds as itself, it can cache the information in various integration profiles. This enables the directory integration server to schedule agents.
- When the directory integration server operates on behalf of an agent, it uses the agent credentials to bind to the directory and perform various operations. The directory integration server can perform only those operations in the directory that are permitted to the agent.

To establish and manage access rights granted to directory integration servers, the Oracle Directory Integration platform creates a group entry, called `odisgroup`, during installation. When a directory integration server is registered, it becomes a member of this group.

You control the access rights granted to directory integration servers by placing access control policies in the `odisgroup` entry. The default policy grants various rights to directory integration servers for accessing the profiles. For example, the default policy enables the directory integration server to compare user passwords for authenticating agents when it binds on their behalf. It also enables directory integration servers to modify status information in the profile—such as the next synchronization time and the synchronization status.

Access Controls for Agents

To control access to Oracle Internet Directory data by agents, place appropriate access control policies in Oracle Internet Directory. This enables you to protect data of one agent from interference by other agents. It also enables you to allow only the agent that owns an attribute to modify that attribute.

To control access, a group entry called `odipgroup` is created in the directory during installation. The access rights granted to various agents in the Oracle Internet Directory Platform are controlled by placing appropriate access policies in the `odipgroup` entry. Each agent is a member of this group. The membership is established when the agent is registered in the system. The default access policy, which is installed automatically with the product, grants various access rights to the agents for the integration profiles they own. For example, the agent can modify the status information such as `orclodipConDirLastAppliedChgTime` in the integration profile. The default access policy also permits agents to access Oracle Internet Directory change logs. The access to the Oracle Internet Directory change log is otherwise restricted.

The `odipgroup` and the `odisgroup` group entries and their default policies are created only during the server installation of the Oracle Internet Directory release 3.0.1 patch. Client-only installations do not create these groups and policies. For this reason, Oracle Corporation recommends that you install the 2.1.1.1 patch on the Oracle Internet Directory release 2.1.1 server. Do this even if you do not intend to use the Oracle Directory Integration platform on the Oracle Internet Directory server installation.

Data Integrity

The Oracle Directory Integration platform ensures that data has not been modified, deleted, or replayed during transmission by using SSL. This SSL feature generates a cryptographically secure message digest—through cryptographic checksums using either the MD5 algorithm or the Secure Hash Algorithm (SHA) —and includes it with each packet sent across the network.

Data Privacy

The Oracle Directory Integration platform ensures that data is not disclosed during transmission by using public-key encryption available with SSL. In public-key encryption, the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the recipient decrypts the message using the recipient's private key.

To exchange data securely between the directory integration server and Oracle Internet Directory, you run both components in the SSL mode.

Tools Security

You can run all the commonly used tools in the SSL mode to transmit data to Oracle Internet Directory securely. These tools include:

- Oracle Directory Manager —Use it to administer data in the directory
- The Oracle directory integration server registration tool (odisrvreg)—Use it to register the directory integration server in the directory
- Ldapadd and ldapmodify tools—Use these to add or modify entries from the command line

26

Bootstrapping a Directory in the Oracle Directory Integration Platform

This chapter contains these topics:

Note: The bootstrapping procedures in this chapter assume that the agent for a connected directory is available to synchronize between the connected directory and Oracle Internet Directory. The procedures are for only the initial synchronization or migration of data from one directory to the other.

Bootstrapping Oracle Internet Directory from a Connected Directory

As part of the initial set up of the Oracle Directory Integration platform environment, you may need to bootstrap Oracle Internet Directory from another directory. You can do this in two ways:

- Using External Tools to Import Data into Oracle Internet Directory
- Using an Agent to Import Data in Oracle Internet Directory

If the directory from which the Oracle Internet Directory is being bootstrapped is also going to be part of the Oracle Directory Integration platform environment—as is the case, for example, with Oracle HR—then follow the steps in this chapter for the initial bootstrap.

Using External Tools to Import Data into Oracle Internet Directory

1. Disallow any updates to the connected directory by setting it to the read-only mode.
2. Copy the data from the connected directory to Oracle Internet Directory by using the external tool. You can do this by using the Oracle Internet Directory bulkload tool, which enables you to load data into Oracle Internet Directory from an LDIF file.

   See Also: "Command-Line Tools Syntax" on page A-4 for instructions on using command-line tools

3. Register the connected directory agent with the Oracle Directory Integration platform by using Oracle Directory Manager. When you do this:
   - Set either the `orclodipConDirLastAppliedChgTime` attribute or the `orclodipConDirLastAppliedChgNum` attribute to the value it had when you set the connected directory to read-only mode
   - Set the `orclodipAgentControl` attribute to `DISABLE`.

   See Also: "Registering the Oracle Directory Integration Server" on page 24-4

4. After copying is complete and verified:
   - Set the connected directory back to the update mode
   - Set the `orclodipAgentControl` attribute in the profile to `ENABLE` so that synchronization between Oracle Internet Directory and the connected directory can start

Using an Agent to Import Data in Oracle Internet Directory

With this method, the agent pulls changes from the connected directory based on a timestamp. For this to happen, the connected directory must identify its changes by using a timestamp.

1. Disallow any updates to the connected directory by setting it to the read-only mode.
2. Register the connected directory agent with the Oracle Directory Integration platform by using Oracle Directory Manager. When you do this:
   - Set either the `orclodipConDirLastAppliedChgTime` attribute or the `orclodipConDirLastAppliedChgNum` attribute to the value it had when you set the connected directory to read-only mode
   - Set the `orclodipAgentControl` attribute to `DISABLE`.

   See Also: "Registering the Oracle Directory Integration Server" on page 24-4

3. Wait for the synchronization to happen. The Oracle directory integration server starts the agent at the scheduled time. The agent then copies the data from the connected directory into Oracle Internet Directory.
4. When copying is complete and verified:
   - Set the connected directory back to the update mode
   - Set the `orclodipAgentControl` attribute in the profile to `ENABLE` so that synchronization between Oracle Internet Directory and the connected directory can start

Bootstrapping a Connected Directory from Oracle Internet Directory

1. Disallow any updates to Oracle Internet Directory by setting it to the read-only mode.
2. Register the connected directory agent with the Oracle Directory Integration platform by using Oracle Directory Manager. When you do this:
   - Set either the `orclodipConDirLastAppliedChgTime` or `orclodipConDirLastAppliedChgNum` attribute in the profile to the value it had when you set the Oracle Internet Directory to read-only mode
   - Set the `orclodipAgentControl` attribute in the profile to `DISABLE`.

   See Also: "Registering the Oracle Directory Integration Server" on page 24-4

3. By using the Oracle Internet Directory ldifwrite tool, move data that is to be synchronized in the connected directory into an LDIF file.
4. Copy data from the LDIF file into the connected directory by using external tools available in the connected directory.
5. When copying is complete and verified:
   - Set Oracle Internet Directory back to the update mode
   - Set the `orclodipAgentControl` attribute in the profile to `ENABLE` so that the synchronization between Oracle Internet Directory and the connected directory can start

27

Synchronizing with Oracle Human Resources

If you store employee data in Oracle Internet Directory, and if you use Oracle Human Resources to create, modify, and delete that data, then you must ensure that the data is synchronized between the two. The Oracle Human Resources agent enables you to do this.

This chapter introduces the Oracle Human Resources agent and explains how to deploy it. It contains these topics:

- Introduction
- Data that You Can Import from Oracle Human Resources
- Managing Synchronization with Oracle Human Resources

Introduction

The Oracle Human Resources agent enables you to import a subset of employee data from Oracle Human Resources into Oracle Internet Directory. It is installed, with a default configuration, along with Oracle Internet Directory. It is ready to run out of the box.

The Oracle Human Resources agent is deployed in the Oracle Directory Integration platform as a partner agent. You can schedule it to run at any time, configuring it to extract incremental changes from the Oracle Human Resources system as often as every second. You can also set and modify attribute mapping between Oracle Human Resources and Oracle Internet Directory.

The Oracle Human Resources agent executable name is `odihragent` and is located in the `$`*`ORACLE_HOME`*`/ldap/dip/bin` directory. You can manage the Oracle Human Resources agent by using Oracle Directory Manager.

Data that You Can Import from Oracle Human Resources

Table 27–1 lists the tables in the Oracle Human Resources schema, most of whose attributes you can import into Oracle Internet Directory:

Table 27–1 Tables in Oracle Human Resources Schema

Table Name	Alias Used in the Agent Config Info Field
PER_PEOPLE_F	PER
PER_ADDRESSES	PA
PER_PERIOD_OF_ SERVICE	PPS
PER_PERSON_TYPE	PPT

All of these tables are visible if the login to the Oracle Human Resources database is done with the apps account.

Because attributes can be added or deleted at runtime from the configuration file, the Oracle Human Resources agent dynamically creates a SQL statement that selects and retrieves only the required attributes.

Table 27–2 shows some of the fields in the Oracle Human Resources user interface. These fields appear when you add or modify employee data.

Table 27–2 Fields in the Oracle Human Resources User Interface

ATTRIBUTE NAME	DESCRIPTION	FORM/CANVAS/FIELD_NAME
LAST_NAME	Last name of the person	People / Name / Last
FIRST_NAME	First name of the person	People / Name / First
TITLE	Title of the person	People / Name / Title
SUFFIX	Suffix—for example, Jr, Sr, Ph.D.	People / Name / Suffix
MIDDLE_NAME	Middle name	People / Name / Suffix
SEX	Sex	Gender List box
START_DATE	Hiring date	People / Hire Date
DATE_OF_BIRTH	Date of birth	People / Personal Information / Birth Date
MARITAL_STATUS	Marital status	People / Personal Information / Status
NATIONAL_ INDENTIFIER	Social security number for US residents	People / Identification / Social Security
EMPLOYEE_NUMBER	Employee number	People / Identification / Employee
REGISTERD_ DISABLED_ FLAG	Indicator that the employee has a disability	People / Personal Information / Has Disability
EMAIL_ADDRESS	Electronic mail address	People / Personal Information / EMail
OFFICE_NUMBER	Office location	People / Office Location Info / Office
MAILSTOP	Mail delivery stop	People / Office Location Info / Mail Stop
INTERNAL_ LOCATION	Location	People / Office Location Info / Location
ADDRESS_LINE1		Personal Address Information / Address line 1
ADDRESS_LINE2		Personal Address Information / Address line 2
ADDRESS_LINE3		Personal Address Information / Address line 3
TOWN_OR_CITY		Personal Address Information / City
REGION_1		Personal Address Information / County

Table 27–2 Fields in the Oracle Human Resources User Interface

ATTRIBUTE NAME	DESCRIPTION	FORM/CANVAS/FIELD_NAME
REGION_2		Personal Address Information / State
POSTAL_CODE		Personal Address Information / Zip Code
COUNTRY		Personal Address Information / Country
TELEPHONE_ NUMBER_1		Personal Address Information / Telephone
TELEPHONE_ NUMBER_2		Personal Address Information / Telephone2

Managing Synchronization with Oracle Human Resources

This section contains these topics:

- Configuring a Directory Integration Profile for the Oracle Human Resources Agent
- Customizing the List of Attributes to Be Synchronized with Oracle Internet Directory
- Customizing Mapping Rules for the Oracle Human Resources Agent
- Running Synchronization from Oracle Human Resources to Oracle Internet Directory

Configuring a Directory Integration Profile for the Oracle Human Resources Agent

To deploy the Oracle Human Resources agent, you must create a directory integration profile for it in Oracle Internet Directory. You can do this by using the procedures outlined in Chapter 24, "Managing the Oracle Directory Integration Server". However, if you have a server installation—that is, a typical installation—then you can use the default integration profile that the Oracle Universal Installer created in the directory for you. A client-only installation does not include this integration profile.

The integration profile contains several attributes and attribute values. Table 27–3 lists these attributes by both their friendly names as used by Oracle Directory Manager—for example, `Agent Name`—and their actual names—for example, `orclodipAgentName`. It provides a description of each attribute, and, where appropriate, the default values in the Oracle Human Resources agent integration profile. Some cells in Table 27–3 contain italicized text providing information and instructions specific to the Oracle Human Resources agent.

Table 27–3 Attributes in the Oracle Human Resources Agent Integration Profile

Attribute	Description
General Information	
Agent Name (`orclodipAgentName`)	Unique name by which the agent is identified in the system. This name is used as an RDN component of the DN that identifies the integration profile. The name can contain only alpha-numeric characters. This attribute is mandatory. You can modify this attribute. *The default name is `OracleHRAgent`. However, if there is already an Oracle Human Resources agent in the system with this name, then you must change the Oracle Human Resources agent name to something else.*
Agent Control (`orclodipAgentControl`)	Indicates whether the agent is enabled or disabled. Valid values are `ENABLE` or `DISABLE`. The default is `DISABLE`. This attribute is mandatory. You can modify this attribute. *You must set this value to `ENABLE`.*
Agent Password (`orclodipAgentPassword`)	This is the password that the directory integration server uses to bind to Oracle Internet Directory on behalf of the agent. This attribute is mandatory. You can modify this attribute. *Set this value to whatever password you want the Oracle Human Resources agent to use.*
Host Name (`orclodipAgentHostName`)	Host on which the agent runs. This attribute is mandatory. You can modify this attribute.
Synchronization Mode (`orclodipSynchronizationMode`)	The direction of synchronization between Oracle Internet Directory and a connected directory ▪ `IMPORT` indicates importing changes from a connected directory to Oracle Internet Directory. ▪ `EXPORT` indicates exporting changes from Oracle Internet Directory to a connected directory. The default is `IMPORT`. This attribute is mandatory. You can modify this attribute. **Note:** Oracle Internet Directory release 3.0.1 support import operations only.

Table 27–3 Attributes in the Oracle Human Resources Agent Integration Profile

Attribute	Description
Scheduling Interval (`orclodipSchedulingInterval`)	Time interval in seconds after which a connected directory is synchronized with Oracle Internet Directory. The default is `600`. This attribute is mandatory. You can modify this attribute.
Number of Retries (`orclodipSyncRetryCount`)	Maximum number of times the directory integration server would try to perform synchronization before disabling it completely. The default is `5`. This attribute is mandatory. You can modify this attribute.
Execution Information	

Agent Execution Command (`orclodipAgentExeCommand`)	Agent executable name and argument list used by the directory integration server to execute the agent.
	This attribute is mandatory. You can modify this attribute.
	The default is:
	`odihragent`
	`connect=hrdb`
	`login=%orclodipConDirAccessAccount`
	`pass=orclodipConDirAccessPassword`
	`date=orclodipConDirLastAppliedChgTime`
	`config=% orclodipAgentConfigInfo`
	`outfile=%s`
	You must set the value in the argument `connect=hrdb` *to the connect string of the Oracle Human Resources system database.*

Connected Directory Account (`orclodipConDirAccessAccount`)	Valid user account in the Oracle Human Resources system that you want to access changes in the Oracle Human Resources system. This information is passed by the directory integration server to the agent in the command line at time of agent's invocation. This attribute is optional. You can modify this attribute.
Connected Directory Account Password (`orclodipConDirAccessPassword`)	Password for the user account accessing the Oracle Human Resources system. It is passed by the directory integration server to the agent at time of agent invocation. This attribute is optional. You can modify this attribute.

Table 27–3 Attributes in the Oracle Human Resources Agent Integration Profile

Attribute	Description
Agent Config Info (`orclodipAgentConfigInfo`)	Any configuration information that you want an agent to store in Oracle Internet Directory. It is passed by the directory integration server to the agent at time of agent invocation. The information is stored as a binary attribute and the directory integration server does not have any knowledge of its content. This is a binary value. The value stored in this attribute represents data to be synchronized from Oracle Human Resources. It is discussed in "Customizing the List of Attributes to Be Synchronized with Oracle Internet Directory" on page 27-9. This attribute is optional. You can modify this attribute.
Datafile Format (`orclodipDatafileType`)	The type of the import or export file. Types are either LDIF or Tagged. The default is `TAGGED`. This attribute is optional. You can modify this attribute.
Mapping Information	
Subscribed Domain (`orclodipChangeSubscriptionDomain`)	DN of the subtree in Oracle Internet Directory to which an agent subscribes for all the changes exported by the agent.
DN Construct Rule (`orclodipEntryDNConstructRule`)	Rule for generating DN for an entry in Oracle Internet Directory from its RDN during an import operation. For example, for `cn=%s, dc=oracle,dc=com`, the %s becomes replaced by the actual RDN value. The default is `cn=%s, dc=Oracle, dc=com` This attribute is optional. You can modify this attribute. *You must change this value to an appropriate DN of the entry under which you want to create the employee entries*
Synchronization Key (`orclodipSynchronizationKey`)	Attribute that uniquely identifies records in a connected directory. This is used as a key to perform synchronization between Oracle Internet Directory and the connected directory. The default is `employeenumber`. This attribute is optional. You can modify this attribute.
Mapping Filter (`orclodipMappingFilter`)	Attribute used to filter the changes in Oracle Internet Directory that are not required for a connected directory.

Table 27–3 Attributes in the Oracle Human Resources Agent Integration Profile

Attribute	Description
Attribute Mapping Rules (`orclodipAttributeMappingRules`)	The mapping rules for mapping data between a connected directory and Oracle Internet Directory, stored as a binary attribute. This is a binary value. The value stored in this attribute is discussed under "Mapping Rules" on page 23-12. This attribute is optional. You can modify this attribute.
Status Information	
Next Synchronization Time (`orclodipNextSynchronizationTime`)	Time when the agent is to be executed next. Its format is `dd-mon-yyyy hh24:mi:ss`. The default is `01-Jan-2001 00:00:00` This attribute is mandatory. You can modify this attribute.
Synchronization Status (`orclodipSynchronizationStatus`)	Execution status of the agent. The default is `0`. This attribute is mandatory. It is read-only.
Synchronization Errors (`orclodipSynchronizationErrors`)	Error message for the last error encountered. This attribute is multivalued. This attribute is mandatory. It is read-only.
Con Dir Last Applied Change Time (`orclodipConDirLastAppliedChgTime`)	Time when the last change from the connected directory was applied to Oracle Internet Directory.Its format is `dd-mon-yyyy hh24:mi:ss`. The default is `01-Jan-2001 00:00:00` This attribute is mandatory. You can modify this attribute.
Con Dir Last Applied Change Num (`orclodipConDirLastAppliedChgNum`)	For agents performing import operations, indicates the last change from the connected directory that has been applied to Oracle Internet Directory.
OID Last Applied Change Number (`orclLastAppliedChangeNumber`)	Time when the last change from Oracle Internet Directory was applied to the local directory.

Customizing the List of Attributes to Be Synchronized with Oracle Internet Directory

You can customize the list of Oracle Human Resources attributes you want to synchronize with Oracle Internet Directory. To help you do this, Oracle Internet Directory includes a default list of Oracle Human Resources attributes to be synchronized. You can modify this list by including additional attributes in it, or excluding some from it.

The default attribute list is stored in the `orclodipAgentConfigInfo` attribute as part of the integration profile. The integration profile is loaded into Oracle Internet Directory as part of a typical installation. The list is also contained in the file named `oraclehragent.cfg.master` and is located under the `$`*`ORACLE_HOME`*`/ldap/odi/conf` directory.

Note: Do not modify the `oraclehragent.cfg.master` file; it serves as a backup.

The columns in the default list of Oracle Human Resources attributes are:

ATTRNAME	The output tag generated in the output data file
COLUMN_NAME	Database column name from where to obtain this value
TABLE_NAME	Database table name from where to obtain this value
FORMAT	The column data type of this attribute. (ASCII, NUMBER, DATE)
MAP	Indicator of whether to extract this attribute from Oracle Human Resources or not. A value of `Y` indicates that it will be extracted and a value of `N` indicates that it will not be.

The `oraclehragent.cfg.master` file contains the following:

```
ATTRNAME:COLUMN_NAME:TABLE_NAME:FORMAT:MAP
PersonId:person_id:PER:NUMBER:Y
PersonType:person_type_id:PER:NUMBER:Y
PersonTypeName:system_person_type:PPT:ASCII:Y
LastName:last_name:PER:ASCII:Y
StartDate:start_date:PER:DATE:Y
BirthDate:date_of_birth:PER:DATE:Y
EMail:email_address:PER:ASCII:Y
EmployeeNumber:employee_number:PER:NUMBER:Y
FirstName:first_name:PER:ASCII:Y
FullName:full_name:PER:ASCII:Y
knownas:known_as:PER:ASCII:Y
MaritalStatus:marital_status:PER:ASCII:Y
middleName:middle_names:PER:ASCII:Y
country:country:PA:ASCII:Y
socialsecurity:national_identifier:PER:ASCII:Y
Sex:sex:PER:ASCII:Y
Title:title:PER:ASCII:Y
suffix:suffix:PER:ASCII:Y
street1:address_line1:PA:ASCII:Y
zip:postal_code:PA:ASCII:Y
Address1:address_line1:PA:ASCII:Y
Address2:address_line2:PA:ASCII:Y
Address3:address_line3:PA:ASCII:Y
TelephoneNumber1:telephone_number_1:PA:ASCII:Y
TelephoneNumber2:telephone_number_2:PA:ASCII:Y
TelephoneNumber3:telephone_number_3:PA:ASCII:Y
town_or_city:town_or_city:PA:ASCII:Y
state:region_2:PA:ASCII:Y
Start_date:effective_start_date:PER:DATE:Y
End_date:effective_end_date:PER:DATE:Y
per_updateTime:last_update_date:PER:DATE:Y
pa_updateTime:last_update_date:PA:DATE:Y
```

Including Additional Oracle Human Resources Attributes for Synchronization

To include additional Oracle Human Resources attributes for synchronization, follow these steps:

1. Copy the `oraclehragent.cfg.master` file and name it anything other than *Agent_Name*.cfg. This is because the directory integration server generates a configuration file with that name, using it to pass the configuration information to the Oracle Human Resources agent at run time.
2. Include an additional Oracle Human Resources attribute for synchronization by adding a record to this file. To do this, you need this information:

   - Table name in the database from which the attribute value is to be extracted. These tables are listed in Table 27–1 on page 27-2. The file uses abbreviated names for the four tables used in the synchronization.
   - Column name in the table
   - Column datatype. Valid values are ASCII, NUMBER, DATE

   You also need to assign an attribute name to the column name. This acts as the output tag by which this attribute is identified in the output file. This tag is also used in the mapping rules to establish a rule between the Oracle Human Resources attribute and the Oracle Internet Directory attribute.

   You must also ensure that the `map` column—that is, the last column in the record—is set to the value Y.

   > **Note:** If you add a new attribute in the attribute list, then you must define a corresponding rule in the `orclodipAttributeMappingRules` attribute. Otherwise the Oracle Human Resources attribute is not synchronized with the Oracle Internet Directory even if it is being extracted by the Oracle Human Resources agent. See "Creating Oracle Human Resources Attribute Mapping Rules" on page 27-14 for instructions about creating mapping rules.

3. Load the file into the `orclodipAgentConfigInfo` attribute by using the ldapmodify tool. The changes take effect the next time the agent runs.

Excluding Oracle Human Resources Attributes from Synchronization

To exclude an Oracle Human Resources attribute that is currently being synchronized with Oracle Internet Directory:

1. Copy the `oraclehragent.cfg.master` file and name it anything other than *Agent_Name*.cfg. This is because the directory integration server generates a configuration file with that name, using it to pass the configuration information to the Oracle Human Resources agent at run time.
2. Do one of the following:
   - Comment out the corresponding record in the attribute list by putting a hash sign (#) in front of it
   - Set the value of the column `map` to `N`
3. Load the file into the `orclodipAgentConfigInfo` attribute by using the ldapmodify tool. The changes take effect the next time the agent runs.

Customizing Mapping Rules for the Oracle Human Resources Agent

Attribute mapping rules govern how the directory integration server converts attributes between Oracle Human Resources and Oracle Internet Directory. You can customize the mapping rules you want the directory integration server to use.

To help you do this, Oracle Internet Directory includes a default list of Oracle Human Resources mapping rules for the Oracle Human Resources system. You configure, modify, and delete mapping rules by editing this list.

The default list of mapping rules is stored in the `orclodipAttributeMappingRules` attribute in the integration profile. In addition, the rules are also in the file named `oraclehragent.map.master` located under the `$`*`ORACLE_HOME`*`/ldap/odi/conf` directory.

Note: Do not modify the `oraclehragent.map.master` file; it serves as a backup.

Default Oracle Human Resources Agent Mapping Rules

The `oraclehragent.map.master` file contains the following:

```
OIDCLASSNAME:OIDATTRIBNAME:OIDATTRIBTYPE:CONNDIRCLASS:CONNDIRATTRIBNAME:CONNDIRATTRIBTYPE:MAPPINGRULE
person:cn: : :1: :copy_string,lastname;append_string,firstname,,
person:sn: : :LastName: :COPY_STRING
#:start_date: ::StartDate::Copy_String_lower
person:rdn: : : : :trim_string,email,@
#organizationalperson:birthday : : :BirthDate: :copy_string
inetOrgperson:mail: : :EMail: :copy_String
inetOrgperson:employeenumber: : :EmployeeNumber: :copy_string
person:cn: : :FirstName: :copy_string
person:cn: : :2 ::copy_string,firstname;append_string,lastname,,
country:c: : :country: :copy_string
#person:ssn: : :socialsecurity: :copy_string
#person:sex: : :Sex: :copy_string
#organizationalperson:title: : :Title: :copy_string
#person:postaladdress: : :Address1: :copy_string
#person:postaladdress: : :Address2: :copy_string
#person:postaladdress: : :Address3: :copy_string
person:telephonenumber: : :TelephoneNumber1: :copy_string
person:telephonenumber: : :TelephoneNumber2: :copy_string
person:telephonenumber: : :TelephoneNumber3: :copy_string
locality:l: : :town_or_city: :copy_string
:changetype: : :changetype: :copy_string
person:userpassword: : : : :literal,welcome
#orclperson:uid: : :EMail: :trim_String,email,@
inetOrgperson:cn: : :EMail: :trim_String,email,@
inetOrgperson:cn: : :lastname: :copy_string
#inetOrgperson:dnqualifier: : :EMail: :copy_string
locality:st: : :state: :copy_string
locality:street: : :street1: :copy_string
locality:postalCode: : :zip: :copy_string
```

The default mapping rules in the `orclodipAttributeMappingRules` attribute correspond to the default Oracle Human Resources attributes list in the `orclodipAgentConfigInfo` attribute. To establish mappings between Oracle Human Resources attributes and Oracle Internet Directory attributes, the mapping rules use the `ATTRNAME` column in each record of the Oracle Human Resources attributes list.

> **See Also:** "Mapping Rules" on page 23-12 for the description of the format of the mapping rules records

Creating Oracle Human Resources Attribute Mapping Rules

To create Oracle Human Resources attribute mapping rules, you modify the `orclodipAttributeMappingRules` attribute. To do this:

1. Copy the `oraclehragent.map.master` file and name it anything other than *`Agent_Name`*`.map`, which is reserved for use by the directory integration server.
2. Add a new rule to this file by adding a record to it. To do this, you need this information:
   - The Oracle Human Resources attribute name that is mapped to Oracle Internet Directory
   - The corresponding attribute in Oracle Internet Directory and its object class to which the Oracle Human Resources attribute are to map
   - The import rule that determines how to map the Oracle Human Resources attribute to the Oracle Internet Directory attribute
3. Load the file into the `orclodipAttributeMappingRules` attribute by using the ldapmodify tool. The changes take effect the next time the agent runs.

Modifying Oracle Human Resources Attribute Mapping Rules

To modify existing Oracle Human Resources attribute mapping rules, you modify the `orclodipAttributeMappingRules` attribute. To do this:

1. Copy the `oraclehragent.map.master` file and name it anything other than *`Agent_Name`*`.map`, which is reserved for use by the directory integration server.
2. Edit this file.
3. Load the file into the `orclodipAttributeMappingRules` attribute by using the ldapmodify tool. The changes take effect the next time the agent runs.

Deleting Oracle Human Resources Attribute Mapping Rules

To delete existing Oracle Human Resources attribute mapping rules, you modify the `orclodipAttributeMappingRules` attribute. To do this:

1. Copy the `oraclehragent.map.master` file and name it anything other than *`Agent_Name`*`.map`, which is reserved for use by the directory integration server.
2. Do one of the following:
   - Delete the rule from the file
   - Comment it out by putting a hash (#) sign in front of it.
3. Load the file into the `orclodipAttributeMappingRules` attribute by using the ldapmodify tool. The changes take effect the next time the agent runs.

Running Synchronization from Oracle Human Resources to Oracle Internet Directory

This section explains how to set up synchronization from Oracle Human Resources to Oracle Internet Directory.

During synchronization, the Oracle Directory Integration platform uses an import file. This file can contain a few or many changes that the Oracle Human Resources agent extracts from the Oracle Human Resources system.

This file is in the tagged format and acts as input to the Oracle directory server. It is named *`Oracle_HR_Agent_Name`*`.data` and is located in `$`*`ORACLE_HOME/`*`ldap/odi/import`.

You do not need to modify this file, but the last version of it is stored in the directory `$`*`ORACLE_HOME/`*`ldap/odi/import/archive` to help you with troubleshooting.

This is an example of an Oracle Human Resources change record in the import file:

```
FirstName: John
LastName: Liu
EmployeeNumber: 12345
Title: Mr.
Sex: M
MaritalStatus: Married
TelephoneNumber: 123-456-7891
Mail: Jliu@my_company.com
Address: 100 Jones Parkway
City: MyTown
```

Preparing for Synchronization

To prepare for synchronization between Oracle Human Resources and Oracle Internet Directory, follow these steps:

1. Ensure that the Oracle Human Resources agent and the directory integration server are installed on the host from which you want to run the Oracle Human Resources agent.

    > **See Also:** The file `install.txt` and the Readme file for Oracle Internet Directory release 3.0.1 for more details

2. Ensure that you have the information for accessing the Oracle Human Resources system, including:
    - Connect string to the Oracle Human Resources system database
    - Access account
    - Password
3. Ensure that the directory integration server on this host is registered in Oracle Internet Directory.

    > **See Also:** “"Registering the Oracle Directory Integration Server" on page 24-4 for registration instructions

4. Configure an integration profile for the Oracle Human Resources agent, as described in "Configuring a Directory Integration Profile for the Oracle Human Resources Agent" on page 27-4. Ensure that all values in the integration profile are properly set, including:
    - Oracle Human Resources attribute list
    - Oracle Human Resources attribute mapping rules
    - Scheduling interval
5. Once everything is properly set, set the `orclodipAgentControl` attribute to `ENABLE`. This indicates that the Oracle Human Resources agent is ready to run.
6. Start the Oracle directory server and the Oracle Human Resources system if they are not already running on the respective hosts.
7. When everything is ready, start the directory integration server if it is not already running on this host.

See Also: "Managing the Oracle Directory Integration Server" on page 24-7 for instructions about starting and stopping the directory integration server

The Synchronization Process

Once the Oracle Human Resources system, Oracle Internet Directory, and the directory integration server are running and the Oracle Human Resources agent is enabled, the directory integration server automatically starts synchronizing changes from the Oracle Human Resources system into Oracle Internet Directory. It follows this process:

1. At the time specified in the `orclodipNextSynchronizationTime` attribute, the directory integration server executes the Oracle Human Resources agent.
2. The Oracle Human Resources agent extracts all the change records from the Oracle Human Resources system based on the time specified in the `orclodipConDirLastAppliedChgTime` attribute in the integration profile. It writes the changes into the Oracle Human Resources import file, namely, `$ORACLE_HOME/ldap/odi/import/`*`HR_Agent_Name`*`.data`. It extracts only the attributes specified in the `orclodipAgentConfigInfo` attribute of the integration profile.
3. Once the Oracle Human Resources agent finishes extracting all the changes, it updates the `orclodipConDirLastAppliedChgTime` attribute to the current time.
4. After the agent completes its execution, the directory integration server updates the changes in Oracle Internet Directory by doing the following:
   - It reads each change record from the import file
   - It converts each change record into an LDAP change entry based on the rules specified in the `orclodipAttributeMappingRules` attribute in the integration profile
5. After the directory integration server finishes updating all the changes, it moves the import file to an archive directory, namely, `$`*`ORACLE_HOME`*`/ldap/odi/import/archive`.

Boostrapping Oracle Internet Directory from Oracle HR

There are two ways to bootstrap Oracle Internet Directory from Oracle HR:

- Use the Oracle Human Resources agent. In the integration profile, set the `orclodipConDirLastAppliedChgTime` to a time before Oracle Human Resources was installed.
- Use external tools to migrate data from Oracle Human Resources into Oracle Internet Directory

See Also: Chapter 26, "Bootstrapping a Directory in the Oracle Directory Integration Platform" for further instructions about initial bootstrapping

Part VIII

Appendixes

This part contains these appendixes:

A

Syntax for LDIF and Command-Line Tools

This appendix provides syntax, usage notes, and examples for **LDAP Data Interchange Format (LDIF)** and LDAP command-line tools. It contains these topics:

LDAP Data Interchange Format (LDIF) Syntax

The standardized file format for directory entries is as follows:

```
dn: distinguished_name
attribute_type: attribute_value
.
.
.
objectClass: object_class_value
.
.
.
```

Property	Value	Description
dn:	*RDN*,*RDN*,*RDN*, ...	Separate RDNs with commas.
attribute:	*attribute_value*	This line repeats for every attribute in the entry, and for every attribute value in multi-valued attributes.
objectClass:	*object_class_ value*	This line repeats for every object class.

The following example shows a file entry for an employee. The first line contains the DN. The lines that follow the DN begin with the mnemonic for an attribute, followed by the value to be associated with that attribute. Note that each entry ends with lines defining the object classes for the entry.

```
dn: cn=Suzie Smith,ou=Server Technology,o=Acme, c=US
cn: Suzie Smith
cn: SuzieS
sn: Smith
email: ssmith@us.Acme.com
telephoneNumber: 69332
photo: /ORACLE_HOME/empdir/photog/ssmith.jpg
objectClass: organizationalPerson
objectClass: person
objectClass: top
```

The next example shows a file entry for an organization:

```
dn: o=Acme,c=US
o: Acme
ou: Financial Applications
objectClass: organization
objectClass: top
```

LDIF Formatting Notes

A list of formatting rules follows. This list is not exhaustive.

- All mandatory attributes belonging to an entry being added must be included with non-null values in the LDIF file.

  > **Tip:** To see the mandatory and optional attribute types for an object class, use Oracle Directory Manager. See "Viewing Properties of Object Classes by Using Oracle Directory Manager" on page 7-9.

- Non-printing characters and tabs are represented in attribute values by base-64 encoding.
- The entries in your file must be separated from each other by a blank line.
- A file must contain at least one entry.
- Lines can be continued to the next line by beginning the continuation line with a space or a tab.
- Add a blank line between separate entries.
- Reference binary files, such as photographs, with the absolute address of the file, preceded by a forward slash ("/").
- The DN contains the full, unique directory address for the object.
- The lines listed after the DN contain both the attributes and their values. DNs and attributes used in the input file must match the existing structure of the DIT. Do not use attributes in the input file that you have not implemented in your DIT.
- Sequence the entries in an LDIF file so that the DIT is created from the top down. If an entry relies on an earlier entry for its DN, make sure that the earlier entry is added before its child entry.
- When you define schema within an LDIF file, insert a white space between the opening parenthesis and the beginning of the text, and between the end of the text and the ending parenthesis.

See Also:

- The various resources listed in "Related Documentation" on page xxxvii for a complete list of LDIF formatting rules
- "Using Globalization Support with LDIF Files" on page 9-3

Command-Line Tools Syntax

This section tells you how to use the following tools:

- ldapadd Syntax
- ldapaddmt Syntax
- ldapbind Syntax
- ldapcompare Syntax
- ldapdelete Syntax
- ldapmoddn Syntax
- ldapmodify Syntax
- ldapmodifymt Syntax
- ldapsearch Syntax

ldapadd Syntax

The ldapadd command-line tool enables you to add entries, their object classes, attributes, and values to the directory. To add attributes to an existing entry, use the ldapmodify command, explained in "ldapmodify Syntax" on page A-15.

> **See Also:** "Adding Configuration Set Entries by Using ldapadd" on page 6-11 for an explanation of using ldapadd to configure a server with an input file

ldapadd uses this syntax:

```
ldapadd [arguments] -f filename
```

where *filename* is the name of an LDIF file written with the specifications explained in the section "LDAP Data Interchange Format (LDIF) Syntax" on page A-2.

The following example adds the entry specified in the LDIF file `my_ldif_file.ldi`:

```
ldapadd -p 389 -h myhost -f my_ldif_file.ldi
```

Optional Arguments	Description
-b	Specifies that you have included binary file names in the file, which are preceded by a forward slash character. The tool retrieves the actual values from the file referenced.
-c	Tells ldapadd to proceed in spite of errors. The errors will be reported. (If you do not use this option, ldapadd stops when it encounters an error.)
-D "*binddn*"	When authenticating to the directory, specifies doing so as the entry specified in *binddn*. Use this with the `-w` *password* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 9, "Managing Globalization Support in the Directory".
-f *filename*	Specifies the input name of the LDIF format import data file. For a detailed explanation of how to format an LDIF file, see "LDAP Data Interchange Format (LDIF) Syntax" on page A-2.
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
-K	Same as `-k`, but performs only the first step of the Kerberos bind
-k	Authenticates using Kerberos authentication instead of simple authentication. To enable this option, you must compile with KERBEROS defined. You must already have a valid ticket granting ticket.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-n	Shows what would occur without actually performing the operation
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *directory_server_port_number*	Connects to the directory on TCP port *directory_server_port_number*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections

Optional Arguments	Description
-U *SSLAuth*	Specifies SSL authentication mode: - 1 for no authentication required - 2 for one way authentication required - 3 for two way authentication required
-v	Specifies verbose mode
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Provides the password required to connect

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapaddmt Syntax

ldapaddmt is like ldapadd: It enables you to add entries, their object classes, attributes, and values to the directory. It is unlike ldapadd in that it supports multiple threads for adding entries concurrently.

While it is processing LDIF entries, ldapaddmt logs errors in the `add.log` file in the current directory.

ldapaddmt uses this syntax:

```
ldapaddmt -T number_of_threads -h host -p port -f filename
```

where *`filename`* is the name of an LDIF file written with the specifications explained in the section "LDAP Data Interchange Format (LDIF) Syntax" on page A-2.

The following example uses five concurrent threads to process the entries in the file `myentries.ldif`.

```
ldapaddmt -T 5 -h node1 -p 3000 -f myentries.ldif
```

Note: Increasing the number of concurrent threads improves the rate at which LDIF entries are created, but consumes more system resources.

Optional Arguments	Description
-b	Specifies that you have included binary file names in the data file, which are preceded by a forward slash character. The tool retrieves the actual values from the file referenced.
-c	Tells the tool to proceed in spite of errors. The errors will be reported. (If you do not use this option, the tool stops when it encounters an error.)
-D "*binddn*"	When authenticating to the directory, specifies doing so as the entry is specified in *binddn*. Use this with the `-w` ***`password`*** option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 9, "Managing Globalization Support in the Directory"
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
-K	Same as -k, but performs only the first step of the kerberos bind
-k	Authenticates using Kerberos authentication instead of simple authentication. To enable this option, you must compile with KERBEROS defined. You must already have a valid ticket granting ticket.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-n	Shows what would occur without actually performing the operation.
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-T	Sets the number of threads for concurrently processing entries

Optional Arguments	Description
-U *SSLAuth*	Specifies SSL Authentication Mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-v	Specifies verbose mode
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Provides the password required to connect

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapbind Syntax

The ldapbind command-line tool enables you to see whether you can authenticate a client to a server.

ldapbind uses this syntax:

```
ldapbind [arguments]
```

Optional Arguments	Description
-D "*binddn*"	When authenticating to the directory, specifies doing so as the entry specified in *binddn*. Use this with the `-w` *password* option.
-E ".*character_set*"	Specifies native character set encoding. See Chapter 9, "Managing Globalization Support in the Directory".
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
-n	Shows what would occur without actually performing the operation

Optional Arguments	Description
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies the wallet password required for one-way or two-way SSL connections
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Provides the password required to connect

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapcompare Syntax

The ldapcompare command-line tool enables you to match attribute values you specify in the command line with the attribute values in the directory entry.

ldapcompare uses this syntax:

```
ldapcompare [arguments]
```

The following example tells you whether `Person Nine`’s title is `associate`.

```
ldapcompare -p 389 -h myhost -b "cn=Person Nine,ou=EuroSInet Suite,o=IMC,c=US"
-a title -v associate
```

Mandatory Arguments	Description
-a *attribute name*	Specifies the attribute on which to perform the compare

Mandatory Arguments	Description
-b "*basedn*"	Specifies the distinguished name of the entry on which to perform the compare
-v *attribute value*	Specifies the attribute value to compare

Optional Arguments	Description
-D *binddn*	When authenticating to the directory, specifies doing so as the entry is specified in *binddn*. Use this with the `-w` *password* option.
-d *debug-level*	Sets the debugging level. See "Setting Debug Logging Levels by Using the OID Control Utility" on page 6-27.
-E "*character_set*"	Specifies native character set encoding. See Chapter 9, "Managing Globalization Support in the Directory".
-f *filename*	Specifies the input filename
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Provides the password required to connect

Optional Arguments	Description

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapdelete Syntax

The ldapdelete command-line tool enables you to remove entire entries from the directory that you specify in the command line.

ldapdelete uses this syntax:

```
ldapdelete [arguments] ["entry_DN" | -f input_filename]
```

Note: If you specify the entry DN, then do not use the `-f` option.

The following example uses port 389 on a host named myhost.

```
ldapdelete -p 389 -h myhost "ou=EuroSInet Suite, o=IMC, c=US"
```

Optional Argument	Description
-D "*binddn*"	When authenticating to the directory, uses a full DN for the *binddn* parameter; typically used with the `-w` *password* option.
-d *debug-level*	Sets the debugging level. See "Setting Debug Logging Levels by Using the OID Control Utility" on page 6-27.
-E "*character_set*"	Specifies native character set encoding. See Chapter 9, "Managing Globalization Support in the Directory".
-f *input_filename*	Specifies the input filename
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
-k	Authenticates using authentication instead of simple authentication. To enable this option, you must compile with Kerberos defined. You must already have a valid ticket granting ticket.

Optional Argument	Description
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-n	Shows what would be done, but doesn't actually delete
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-v	Specifies verbose mode
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Provides the password required to connect.

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapmoddn Syntax

The ldapmoddn command-line tool enables you to modify the DN or RDN of an entry.

ldapmoddn uses this syntax:

```
ldapmoddn [arguments]
```

The following example uses ldapmoddn to modify the RDN component of a DN from `"cn=mary smith"` to `"cn=mary jones"`. It uses port 389, and a host named `myhost`.

```
ldapmoddn -p 389 -h myhost -b "cn=mary smith,dc=Americas,dc=imc,dc=com" -R
"cn=mary jones"
```

Mandatory Argument	Description
-b "*basedn*"	Specifies DN of the entry to be moved

Optional Argument	Description
-D "*binddn*"	When authenticating to the directory, do so as the entry is specified in *binddn*. Use this with the `-w` *password* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 9, "Managing Globalization Support in the Directory".
-f *filename*	Specifies the input filename
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-N *newparent*	Specifies new parent of the RDN
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).

Optional Argument	Description
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-r	Specifies that the old RDN is not retained as a value in the modified entry. If this argument is not included, the old RDN is retained as an attribute in the modified entry.
-R *newrdn*	Specifies new RDN
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Provides the password required to connect.

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapmodify Syntax

The ldapmodify tool enables you to act on attributes.

ldapmodify uses this syntax:

```
ldapmodify [arguments] -f filename
```

where *filename* is the name of an LDIF file written with the specifications explained the section "LDAP Data Interchange Format (LDIF) Syntax" on page A-2.

The list of arguments in the following table is not exhaustive.

Optional Argument	Description
-a	Denotes that entries are to be added, and that the input file is in LDIF format.
-b	Specifies that you have included binary file names in the data file, which are preceded by a forward slash character.
-c	Tells ldapmodify to proceed in spite of errors. The errors will be reported. (If you do not use this option, ldapmodify stops when it encounters an error.)
-D "*binddn*"	When authenticating to the directory, specifies doing so as the entry is specified in *binddn*. Use this with the `-w` *password* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 9, "Managing Globalization Support in the Directory".
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-n	Shows what would occur without actually performing the operation.
`-o` *`log_file_name`*	Can be used with the `-c` option to write the erroneous LDIF entries in the logfile. You must specify the absolute path for the log file name.
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).

Optional Argument	Description
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-v	Specifies verbose mode
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Overrides the default, unauthenticated, null bind. To force authentication, use this option with the `-D` option.

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

To run `modify`, `delete`, and `modifyrdn` operations using the `-f` flag, use LDIF for the input file format (see "LDAP Data Interchange Format (LDIF) Syntax" on page A-2) with the specifications noted below:

If you are making several modifications, then, between each modification you enter, add a line that contains a hyphen (-) only. For example:

```
dn: cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
changetype: modify
add: work-phone
work-phone: 510/506-7000
work-phone: 510/506-7001
-
delete: home-fax
```

Unnecessary space characters in the LDIF input file, such as a space at the end of an attribute value, will cause the LDAP operations to fail.

Line 1: Every change record has, as its first line, the literal `dn:` followed by the DN value for the entry, for example:

```
dn:cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
```

Line 2: Every change record has, as its second line, the literal `changetype:` followed by the type of change (`add, delete, modify, modrdn`), for example:

```
changetype: modify
```

or

```
changetype: modrdn
```

Format the remainder of each record according to the following requirements for each type of change:

- `changetype: add`

  Uses LDIF format (see "LDAP Data Interchange Format (LDIF) Syntax" on page A-2).

- `changetype: modify`

  The lines that follow this changetype consist of changes to attributes belonging to the entry that you identified in Line 1 above. You can specify three different types of attribute modifications—add, delete, and replace—which are explained next:

  – **Add attribute values**. This option to changetype modify adds more values to an existing multi-valued attribute. If the attribute does not exist, it adds the new attribute with the specified values:

    ```
    add: attribute name
    attribute name: value1
    attribute name: value2...
    ```

    For example:

    ```
    dn:cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
    changetype: modify
    add: work-phone
    work-phone: 510/506-7000
    work-phone: 510/506-7001
    ```

- **Delete values**. If you supply only the *delete* line, all the values for the specified attribute are deleted. Otherwise, if you specify an attribute line, you can delete specific values from the attribute:

  ```
  delete: attribute name
  [attribute name: value1]
  ```

  For example:

  ```
  dn: cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
  changetype: modify
  delete: home-fax
  ```

- **Replace values.** Use this option to replace all the values belonging to an attribute with the new, specified set:

  ```
  replace: attribute name
  [attribute name: value1 ...]
  ```

  If you do not provide any attributes with `replace`, then the directory adds an empty set. It then interprets the empty set as a delete request, and complies by deleting the attribute from the entry. This is useful if you want to delete attributes that may or may not exist.

  For example:

  ```
  dn: cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
  changetype: modify
  replace: work-phone
  work-phone: 510/506-7002
  ```

  * `changetype:delete`

    This change type deletes entries. It requires no further input, since you identified the entry in Line 1 and specified a changetype of delete in Line 2.

    For example:

    ```
    dn: cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
    changetype: delete
    ```

  * `changetype:modrdn`

    The line following the change type provides the new relative distinguished name using this format:

    ```
    newrdn: RDN
    ```

For example:

```
dn: cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
changetype: modrdn
newrdn: cn=Barbara Fritchy-Blomberg
```

Example: Using ldapmodify to Add an Attribute

This example adds a new attribute called myAttr. The LDIF file for this operation is:

```
dn: cn=subschemasubentry
changetype: modify
add: attributetypes
attributetypes: (1.2.3.4.5.6.7 NAME ‘myAttr’ DESC ‘New attribute definition’
EQUALITY caseIgnoreMatch SYNTAX
‘1.3.6.1.4.1.1466.115.121.1.15’ )
```

On the first line, enter the DN specifying where this new attribute is to be located. All attributes and object classes they are stored in cn=subschemasubentry.

The second and third lines show the proper format for adding a new attribute.

The last line is the attribute definition itself. The first part of this is the object identifier number: 1.2.3.4.5.6.7. It must be unique among all other object classes and attributes. Next is the NAME of the attribute. In this case the attribute NAME is myAttr. It must be surrounded by single quotes. Next is a description of the attribute. Enter whatever description you want between single quotes. At the end of this attribute definition in this example are optional formatting rules to the attribute. In this case we are adding a matching rule of EQUALITY caseIgnoreMatch and a SYNTAX of Directory String. This example uses the object ID number of 1.3.6.1.4.1.1466.115.121.1.15 instead of the SYNTAXES name which is "Directory String".

Put your attribute information in a file formatted like this example. Then run the following command to add the attribute to the schema of your Oracle directory server.

```
ldapmodify -h yourhostname -p 389 -D "orcladmin" -w "welcome" -v -f
/tmp/newattr.ldif
```

This ldapmodify command assumes that your Oracle directory server is running on port 389, that your super user account name is orcladmin, that your super user password is `welcome` and that the name of your LDIF file is `newattr.ldif`. Substitute the host name of your computer where you see *`yourhostname`*.

If you are not in the directory where the LDIF file is located, then you must enter the full directory path to the file at the end of your command. This example assumes that your LDIF file is located in the `/tmp` directory.

ldapmodifymt Syntax

The ldapmodifymt command-line tool enables you to modify several entries concurrently.

ldapmodifymt uses this syntax:

```
ldapmodifymt -T number_of_threads [arguments] -f filename
```

where *`filename`* is the name of an LDIF file written with the specifications explained the section "LDAP Data Interchange Format (LDIF) Syntax" on page A-2.

See Also: "ldapmodify Syntax" on page A-15 for additional formatting specifications used by ldapmodifymt

The following example uses five concurrent threads to modify the entries in the file `myentries.ldif`.

```
ldapmodifymt -T 5 -h node1 -p 3000 -f myentries.ldif
```

Note: The ldapmodifymt tool logs error messages in the file `add.log`, which is located in the directory where you are running the command.

Optional Argument	Description
-a	Denotes that entries are to be added, and that the input file is in LDIF format. (If you are running ldapadd, this flag is not required.)
-b	Specifies that you have included binary file names in the data file, which are preceded by a forward slash character.
-c	Tells ldapmodify to proceed in spite of errors. The errors will be reported. (If you do not use this option, ldapmodify stops when it encounters an error.)

Optional Argument	Description
-D "*binddn*"	When authenticating to the directory, specifies doing so as the entry is specified in binddn. Use this with the `-w` *password* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 9, "Managing Globalization Support in the Directory".
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-n	Shows what would occur without actually performing the operation.
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-T	Sets the number of threads for concurrently processing entries
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-v	Specifies verbose mode
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Overrides the default, unauthenticated, null bind. To force authentication, use this option with the -D option.

Optional Argument	Description

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapsearch Syntax

The ldapsearch command-line tool enables you to search for and retrieve specific entries in the directory.

ldapsearch uses this syntax:

```
ldapsearch [arguments] filter [attributes]
```

The *filter* format must be compliant with RFC-2254.

> **See Also:** http://www.ietf.org/rfc/rfc2254.txt for further information about the standard for the filter format

Separate attributes with a space. If you do not list any attributes, all attributes are retrieved.

Mandatory Argument	Description
-b "*basedn*"	Specifies the base DN for the search
-s *scope*	Specifies search scope: base, one, or sub

Optional Argument	Description
-A	Retrieves attribute names only (no values)
-a *deref*	Specifies alias dereferencing: never, always, search, or find
-B	Allows printing of non-ASCII values
-D "*binddn*"	When authenticating to the directory, specifies doing so as the entry specified in *binddn*. Use this with the `-w` *password* option.

Optional Argument	Description
-d *debug level*	Sets debugging level to the level specified (see Table 6–1 on page 6-28)
-E "*character_set*"	Specifies native character set encoding. See Chapter 9, "Managing Globalization Support in the Directory".
-f *file*	Performs sequence of searches listed in *file*
-F sep	Prints 'sep' instead of '=' between attribute names and values
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
-L	Prints entries in LDIF format (-B is implied)
-l *timelimit*	Specifies maximum time (in seconds) to wait for ldapsearch command to complete
-n	Shows what would be done without actually searching
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-S *attr*	Sorts the results by attribute *attr*
-t	Writes to files in /tmp
-u	Includes user friendly entry names in the output
-U *SSLAuth*	Specifies the SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-v	Specifies verbose mode
-w *passwd*	Specifies bind passwd for simple authentication

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

Optional Argument	Description
-z *sizelimit*	Specifies maximum number of entries to retrieve

Examples of ldapsearch Filters

Study the following examples to see how to build your own search commands.

Example 1: Base Object Search The following example performs a base-level search on the directory from the root.

```
ldapsearch -p 389 -h myhost -b "" -s base -v "objectclass=*"
```

- `-b` specifies base DN for the search, root in this case.
- `-s` specifies whether the search is a base search (`base`), one level search (`one`) or subtree search (`sub`).
- `"objectclass=*"` specifies the filter for search.

Example 2: One-Level Search The following example performs a one level search starting at `"ou=HR, ou=Americas, o=IMC, c=US"`.

```
ldapsearch -p 389 -h myhost -b "ou=HR, ou=Americas, o=IMC, c=US" -s one -v
"objectclass=*"
```

Example 3: Subtree Search The following example performs a subtree search and returns all entries having a DN starting with `"cn=us"`.

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub -v "cn=Person*"
```

Example 4: Search Using Size Limit The following example actually retrieves only two entries, even if there are more than two matches.

```
ldapsearch -h myhost -p 389 -z 2 -b "ou=Benefits,ou=HR,ou=Americas,o=IMC,c=US"
-s one "objectclass=*"
```

Example 5: Search with Required Attributes The following example returns only the DN attribute values of the matching entries:

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub -v "objectclass=*" dn
```

The following example retrieves only the distinguished name along with the surname (`sn`) and description (`description`) attribute values:

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub -v "cn=Person*" dn sn description
```

Example 6: Search for Entries with Attribute Options The following example retrieves entries with common name (cn) attributes that have an option specifying a language code attribute option. This particular example retrieves entries in which the common names are in French and begin with the letter R.

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub "cn;lang-fr=R*"
```

Suppose that, in the entry for John, no value is set for the cn;lang-it language code attribute option. In this case, the following example does not return John's entry:

```
ldapsearch -p 389 -h myhost -b "c=us" -s sub "cn;lang-it=Giovanni"
```

Example 7: Searching for All User Attributes and Specified Operational Attributes The following example retrieves all user attributes and the createtimestamp and orclguid operational attributes:

```
ldapsearch -p 389 -h myhost -b "ou=Benefits,ou=HR,ou=Americas,o=IMC,c=US" -s sub
"cn=Person*" * createtimestamp orclguid
```

The following example retrieves entries modified by Anne Smith:

```
ldapsearch -h sun1 -b "" "(&(objectclass=*)(modifiersname=cn=Anne
Smith))"
```

The following example retrieves entries modified between 01 April 2001 and 06 April 2001:

```
ldapsearch -h sun1 -b "" "(&(objectclass=*)(modifytimestamp >= 20000401000000)
(modifytimestamp <= 20000406235959))"
```

Note: Because modifiersname and modifytimestamp are not indexed attributes, use catalog.sh to index these two attributes. Then, restart the Oracle directory server before issuing the two previous ldapsearch commands.

Other Examples: Each of the following examples searches on port 389 of host sun1, and searches the whole subtree starting from the DN "ou=hr,o=acme,c=us".

The following example searches for all entries with any value for the objectclass attribute.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree "objectclass=*"
```

The following example searches for all entries that have orcl at the beginning of the value for the objectclass attribute.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"objectclass=orcl*"
```

The following example searches for entries where the objectclass attribute begins with orcl and cn begins with foo.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"(&(objectclass=orcl*)(cn=foo*))"
```

The following example searches for entries in which the common name (cn) is not foo.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree "(!(cn=foo))"
```

The following example searches for entries in which cn begins with foo or sn begins with bar.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"(|(cn=foo*)(sn=bar*))"
```

The following example searches for entries in which employeenumber is less than or equal to 10000.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"employeenumber<=10000"
```

Bulk Tools Syntax

This section contains these topics:

- bulkdelete Syntax
- bulkload Syntax
- bulkmodify Syntax
- ldifwrite Syntax

bulkdelete Syntax

The bulkdelete command-line tool enables you to delete a subtree efficiently. It can be used when both an Oracle directory server and Oracle directory replication servers are in operation. It uses a SQL interface to benefit performance. For this release, the bulkdelete tool runs on only one node at a time.

This tool does not support filter-based deletion. That is, it deletes an entire subtree below the root of the subtree. If the base DN is a user-added DN, rather than a DN created as part of the installation of the directory, it is included in the delete. You must restrict LDAP activity against the subtree during deletion.

The bulkdelete tool uses this syntax:

```
bulkdelete.sh -connect net_service_name -base "base_dn" -size number_of_entries
-encode "character_set"
```

Mandatory Argument	Description
- connect *net_service_name*	Specifies the net service name to connect to the directory database **See Also:** *Oracle Net Services Administrator's Guide*
- base "*base_dn*"	Specifies the base DN of the subtree to be deleted

Optional Argument	Description
-size *number_of_entries*	Specifies the number of entries to be committed as a part of one transaction.
-encode "*character_set*"	Native character set encoding

bulkload Syntax

The bulkload command-line tool uses Oracle SQL*Loader to create directory entries from data residing in or created by other applications. When using bulkload, you specify any options and the input filename. Bulkload expects an empty directory and will either fail or overwrite if there are existing entries.The bulkload tool expects the input file to be in LDIF.

> **See Also:** "LDAP Data Interchange Format (LDIF) Syntax" on page A-2.

The bulkload tool uses this syntax:

```
bulkload.sh -connect net_service_name [-check] [-generate] [-load]
   [-restore] absolute_path_to_ldif.file
```

Mandatory Argument	Description
connect *net_service_name*	Specifies the net service name defined in the `tnsnames.ora` file. **See Also:** *Oracle Net Services Administrator's Guide*

Optional Argument	Description
-check	Checks LDAP schema for inconsistencies and for existence of duplicate DNs in the file
-encode "*character_set*"	Specifies native character set encoding. See Chapter 9, "Managing Globalization Support in the Directory".
-generate	Creates files suitable for loading into Oracle Internet Directory
-load	Loads files resulting from generate phase into specified database
-restore	Takes the operational attributes, such as `orclguid`, `creatorsname`, and `createtimestamp`, from the LDIF file rather than generating new ones. Use this argument only when the LDIF file contains operational attributes. Use this in conjunction with the `generate` and `check` arguments.

Bulk loading must be performed when directory server instances are not running.

> **See Also:** Chapter 6, "Managing the Oracle Directory Server" for instructions on stopping directory server instances

The LDIF data file path must be fully specified for check or generate operations.

Bulk Loading Multiple Nodes in a Replicated Environment

After generating a file with the `generate` option, you can use the `load` option to load multiple computers with the identical SQL*Loader file. Do this only when creating a new replica node.

> **See Also:** "Task 7: Start the Replication Servers on All the Nodes" on page 15-19

The current version of bulkload does not allow you to specify the connection information for all of the nodes in one command.

When you load the same data into multiple nodes in a replicated network, ensure that the `orclGUID` parameter (global IDs) is consistent across all the nodes. You can accomplish this by generating the bulkload data file once only (using the `-generate` option), and then using the same data file to load the other nodes (using the `-load` option).

bulkmodify Syntax

The bulkmodify command-line tool enables you to modify a large number of existing entries in an efficient way. The bulkmodify tool supports the following:

- Subtree based modification
- A single attribute filter. For example, the filter could be `objectclass=*`, `objectclass=oneclass`, or `telephonenumber=*`.
- Attribute value addition and replacement. It modifies all matched entries in bulk.

The bulkmodify tool performs schema checking on the specified attribute name and value pair during initialization. All entries that meet the following criteria are modified:

- They are under the specified subtree.
- They meet the single filter condition.
- They contain the attribute to be modified as either mandatory or optional.

The Oracle directory server and Oracle directory replication server may be running concurrently while bulk modification is in progress, but the bulk modification does not affect the replication server. You must perform bulk modification against all replicas.

> **Note:** LDIF file based modification is not supported by bulkmodify. This type of modification requires per entry based schema checking, and therefore the performance gain over the existing ldapmodify tool is insignificant.

You must restrict user access to the subtree during bulk modification. If necessary, **ACI** restriction can be applied to the subtree being updated by bulkmodify.

You cannot use bulkmodify to add a value to single-valued attributes that already contain one value. If a second value is added, you must alter the directory schema to make that attribute multi-valued.

The bulkmodify tool uses this syntax:

```
bulkmodify -c net_service_name -b "base_dn" {-a|-r} attr_name -v att_value [-f
filter] [-s size]
```

Mandatory Argument	Description
-c *net_service_name*	Specifies the net service name of the directory database **See Also:** *Oracle Net Services Administrator's Guide*
-b "*base_dn*"	Specifies the base DN of the subtree to be modified
-a *attr_name*	Specifies the attribute name for addition
-r *attr_name*	Specifies the attribute name for replacement
-v *att_value*	Specifies the attribute value for either addition or replacement

Optional Argument	Description
-f *filter*	Specifies the filter to be used
-s *number_of_entries*	Specifies the number of entries to be committed as a part of one transaction. If not specified, default is 100.
-E "*character_set*"	Specifies native character set encoding. See Chapter 9, "Managing Globalization Support in the Directory".

The filter specified with the `-f` option must contain a single attribute.

If a filter is not specified, the default filter `objectclass=*` is assumed.

There can be only one attribute name specified in the -a or `-r` option in each execution.

There can be only one value specified in the `-v` option in each execution. For example, the following bulkmodify command adds the telephone number 408-123-4567 to the entries of all employees who have Anne Smith as their manager:

```
bulkmodify -c my_database -b "c=US" -a telephoneNumber -v "408-123-4567" -f
"manager=Anne Smith"
```

To assure that the modified entries are read, after completing the bulkmodify procedure, restart the Oracle Internet Directory server.

ldifwrite Syntax

The ldifwrite command-line tool enables you to convert to LDIF all or part of the information residing in an Oracle Internet Directory. This makes that information available for loading into a new node in a replicated directory or into another node for backup storage.

Note: The ldifwrite tool output does not include operational data of the directory itself—for example, `cn=subschemasubentry`, `cn=catalogs`, and `cn=changelog entries`. To export these entries into LDIF format, use ldapsearch with the `-L` flag.

The ldifwrite tool performs a subtree search, including all entries below the specified DN, including the DN itself.

The ldifwrite tool uses this syntax:

```
ldifwrite -c net_service_name -b "base_DN" -f filename
```

Mandatory Argument	Description
-c *net_service_name*	Specifies the net service name of the directory that is the source of the data, as defined in the `tnsnames.ora` file. **See Also:** *Oracle Net Services Administrator's Guide*
-b "*base_dn*"	Specifies the base of the subtree to be written out in LDIF format
-f *filename*	Specifies the name of the LDIF file to be created

Optional Argument	Description
-E "*character_set*"	Specifies native character set encoding. **See Also:** "Using Globalization Support with ldifwrite" on page 9-9

The following example writes all the entries under `ou=Europe, o=imc, c=us` into the `output1.ldi` file.

```
ldifwrite -c nldap -b "ou=Europe, o=imc, c=us" -f output1.ldi
```

All the arguments are mandatory.

The LDIF file and the intermediate file are always written to the current directory.

The ldifwrite tool includes the operational attributes of each entry in the directory, including `createtimestamp`, `creatorsname`, and `orclguid`.

Catalog Management Tool Syntax

Oracle Internet Directory uses indexes to make attributes available for searches. When Oracle Internet Directory is installed, the entry `cn=catalogs` lists available attributes that can be used in a search. Only those attributes that have an equality matching rule can be indexed.

If you want to use additional attributes in search filters, you must add them to the catalog entry. You can do this at the time you create the attribute by using Oracle Directory Manager. However, if the attribute already exists, then you can index it only by using the Catalog Management tool.

The Catalog Management tool uses this syntax:

```
catalog.sh -connect net_service_name {add|delete} {-attr attr_name|-file
filename}
```

Mandatory Argument	Description
- connect *net_service_name*	Specifies the net service name to connect to the directory database **See Also:** *Oracle Net Services Administrator's Guide*

Optional Argument	Description
- add -attr *attr_name*	Indexes the specified attribute
- delete -attr *attr_name*	Drops the index from the specified attribute
- add -file *filename*	Indexes attributes (one per line) in the specified file
-delete -file *filename*	Drops the indexes from the attributes in the specified file

When you enter the `catalog.sh` command, the following message appears:

```
This tool can only be executed if you know the OiD user password.
Enter OiD password:
```

If you enter the correct password, the command is executed. If you give an incorrect password, the following message is displayed:

```
Cannot execute this tool
```

To effect the changes after running the Catalog Management tool, stop, then restart, the Oracle directory server.

> **See Also:** "OID Control Utility Syntax" on page A-35 and for instructions on starting and restarting directory servers. Note that OID Monitor must be running before you start a directory server. See "OID Monitor Syntax" on page A-34 for information about starting OID Monitor.

OID Monitor Syntax

This section contains these topics:

- Starting the OID Monitor
- Stopping the OID Monitor

Starting the OID Monitor

To start the OID Monitor:

1. Set the following environment variable to the appropriate language setting. The default language set at installation is AMERICAN_AMERICA.

   ```
   NLS_LANG=APPROPRIATE_LANGUAGE.UTF8
   ```

2. At the system prompt, type:

   ```
   oidmon [connect=net_service_name] [sleep=seconds] start
   ```

Argument	Description
connect=*net_service_name*	Specifies the net service name of the database to which you want to connect. This is the network service name set in the `tnsnames.ora` file. This argument is optional.
sleep=*seconds*	Specifies number of seconds after which the OID Monitor should check for new requests from OID Control and for requests to restart any servers that may have stopped. The default sleep time is 10 seconds. This argument is optional.
start	Starts the OID Monitor process

For example:

```
oidmon connect=dbs1 sleep=10 start
```

Stopping the OID Monitor

To stop the OID Monitor daemon, at the system prompt, type:

```
oidmon [connect=net_service_name] stop
```

Argument	Description
connect=*net_service_name*	Specifies net service name of the database to which you want to connect. This is the net service name set in the `tnsnames.ora` file.
stop	Stops the OID Monitor process

For example:

```
oidmon connect=dbs1 stop
```

OID Control Utility Syntax

> **Note:** OID Monitor must be running whenever you start, stop, or restart directory server instances.

This section contains these topics:

- Starting and Stopping an Oracle Directory Server Instance
- Starting and Stopping an Oracle Directory Replication Server Instance
- Restarting Directory Server Instances
- Troubleshooting Directory Server Instance Startup

Starting and Stopping an Oracle Directory Server Instance

Use the **OID Control Utility** to start and stop Oracle directory server instances.

Starting an Oracle Directory Server Instance

The syntax for starting an Oracle directory server instance is:

```
oidctl connect=net_service_name server=oidldapd instance=server_instance_number
[configset=configset_number] [flags='-p port_number -work maximum_number_of_
worker_threads_per_server -server number_of_server_processes -debug debug_level
-l change-logging -server n'] start
```

Argument	Description
connect=*net_service_name*	If you already have a tnsnames.ora file configured, this is the net service name specified in that file, located in ORACLE_HOME/network/admin
server=oidldapd	Type of server to start (valid values are OIDLDAPD and OIDREPLD). This is not case-sensitive.
instance=*server_instance_number*	Instance number of the server to start. Should be a number between 0 and 1000.
configset=*configset_number*	Configset number used to start the server. This defaults to configset0 if not set. This should be a number between 0 and 1000.
-p *port_number*	Specifies a port number during server instance startup. Default port if not set is 389.
-work *maximum_number_of_worker_threads_per_server*	Specifies the maximum number of worker threads for this server
-debug *debug_level*	Specifies a debug level during Oracle directory server instance startup
-l *change_logging*	Turns replication change-logging on and off. To turn it off, enter -l. To turn it on, omit the flag. The default is true (values = true and false). (directory server only)
-server *n*	Specifies the number of server processes to start on this port
start	Starts the server specified in the server argument.

For example, to start an Oracle directory server instance whose net service name is dbs1, using configset5,at port 12000, with a debug level of 1024, an instance number 3, and in which change-logging is turned off, type at the system prompt:

```
oidctl connect=dbs1 server=oidldapd instance=3 configset=5 flags='-p 12000
-debug 1024 -l' start
```

When starting and stopping an Oracle directory server instance, the server name and instance number are mandatory. All other arguments are optional.

All keyword value pairs within the flags arguments must be separated by a single space.

Single quotes are mandatory around the flags.

The configset identifier defaults to zero (`configset0`) if not set.

Note: If you choose to use a port other than the default port (389 for non-secure usage or 636 for secure usage), you must tell the clients which port to use to locate the Oracle Internet Directory. If you use the default ports, clients can connect to the Oracle Internet Directory without referencing a port in their connect requests.

Stopping an Oracle Directory Server Instance

At the system prompt, type:

```
oidctl connect=net_service_name server=oidldapd instance=server_instance_number
stop
```

For example:

```
oidctl connect=dbs1 server=oidldapd instance=3 stop
```

Starting and Stopping an Oracle Directory Replication Server Instance

Use the OID Control Utility to start and stop Oracle directory replication server instances.

Starting an Oracle Directory Replication Server Instance

The syntax for starting the Oracle directory replication server is:

```
oidctl connect=net_service_name server=oidrepld instance=server_instance_number
[configset=configset_number] flags='-h hostname -p port_number
-d debug_level -z transaction_size' start
```

Argument	Description
`connect`	If you already have a `tnsnames.ora` file configured, then this is the name specified in that file, which is located in ORACLE_ HOME/network/admin
`server`	Type of server to start (valid values are OIDLDAPD and OIDREPLD). This is not case-sensitive.
`instance`	Instance number of the server to start. Should be a number between 0 and 1000.
`configset`	Configset number used to start the server. This defaults to configset0 if not set. This should be a number between 0 and 1000.
`-p`	Specifies a port number during server instance startup. Default port if not set is 389.
`-d`	Specifies a debug level during replication server instance startup
`-h`	Specifies the host name on which the server runs. (Replication server only)
`-m [true|false]`	Turns conflict resolution on and off. The default is true (values = true and false). (Replication server only)
`-z`	Specifies the number of changes applied in each replication update cycle. If you do not specify this, the number is determined by the Oracle directory server sizelimit parameter, which has a default setting of 1024. You can configure this latter setting.
`start`	Starts the server specified in the *server* argument.

For example, to start the replication server with an instance=1, at port 12000, with debugging set to 1024, type at the system prompt:

```
oidctl connect=dbs1 server=oidrepld instance=1 flags='-p 12000 -h eastsun11 -d
1024' start
```

When starting and stopping an Oracle directory replication server, the `-h` flag, which specifies the host name, is mandatory. All other flags are optional.

All keyword value pairs within the flags arguments must be separated by a single space.

Single quotes are mandatory around the flags.

The configset identifier defaults to zero (`configset0`) if not set.

Note: If you choose to use a port other than the default port (389 for non-secure usage or 636 for secure usage), you must tell the clients which port to use to locate the Oracle Internet Directory. If you use the default ports, clients can connect to the Oracle Internet Directory without referencing a port in their connect requests.

Stopping an Oracle Directory Replication Server Instance

At the system prompt, type:

```
oidctl connect=net_service_name server=oidrepld instance=server_instance_number
stop
```

For example:

```
oidctl connect=dbs1 server=oidrepld instance=1 stop
```

Restarting Directory Server Instances

To restart a directory server instance, at the system prompt, type:

```
oidctl connect=net_service_name server={oidldapd|oidrepld}
instance=server_instance_number  restart
```

OID Monitor must be running whenever you start, stop, or restart directory server instances.

If you try to contact a server that is down, you receive from the SDK the error message `81—LDAP_SERVER_DOWN`.

If you change a configuration set entry that is referenced by an active server instance, you must stop that instance and restart it to effect the changed value in the configuration set entry on that server instance. You can either issue the STOP command followed by the START command, or you can use the RESTART command. RESTART both stops and restarts the server instance.

For example, suppose that Oracle directory server instance1 is started, using configset3, and with the net service name dbs1. Further, suppose that, while instance1 is running, you change one of the attributes in configset3. To enable the change in configset3 to take effect on instance1, you enter the following command:

```
oidctl connect=dbs1 server=oidldapd instance=1 restart
```

If there are more than one instance of the Oracle directory server running on that node using configset3, then you can restart all the instances at once by using the following command syntax:

```
oidctl connect=dbs1 server=oidldapd restart
```

Note that this command restarts all the instances running on the node, whether they are using configset3 or not.

Important Note: During the restart process, clients cannot access the Oracle directory server instance. However, the process takes only a few seconds to execute.

Troubleshooting Directory Server Instance Startup

If the directory server fails to start, you can override all user-specified configuration parameters to start the directory server and then return the configuration sets to a workable state by using the ldapmodify operation.

To start the directory server by using its hard-coded default parameters instead of the configuration parameters stored in the directory, type at the system prompt:

```
oidctl connect=net_service_name flags='-p port_number -f'
```

The `-f` option in the flags starts the server with hard-coded configuration values, overriding any defined configuration sets except for the values in `configset0`.

To see debug log files generated by the OID Control Utility, navigate to `$`*`ORACLE_HOME/`*`ldap/log`.

OID Database Password Utility Syntax

The OID Database Password Utility syntax is:

```
oidpasswd [connect=net_service_name]
```

The OID Database Password Utility prompts you for the current password. Type the current password, then the new password, then a confirmation of the new password.

The OID Database Password Utility assumes by default that the password being changed is that of the local database (as defined by *ORACLE_HOME* and ORACLE_*SID*). If you are changing the password on a remote database, you must use the `connect=`*`net_service_name`* option.

For example:

```
$ oidpasswd
current password: ods
new password: newsupersecret
confirm password: newsupersecret
password set.
$
```

Note: User responses are not echoed to the screen.

Human Intervention Queue Manipulation Tool Syntax

The Human Intervention Queue Manipulation Tool enables you to move the changes from the human intervention queue to either the retry queue or the purge queue. Moving the change to the purge queue means that there are no further attempts to re-apply the changelog entry. Perform the following general steps to address changes in the human intervention queue:

1. Shutdown the Oracle directory replication server.
2. Analyze the replication log.
3. Use the Human Intervention Queue Manipulation Tool to move the changes to either the retry queue or the purge queue as described in the following sections.

Moving a Change from the Human Intervention Queue into the Retry Queue

To place a change back into the retry queue, use this syntax:

```
hiqretry.sh –connect net_service_name [-start change_number]
[-end change_number] [-equal change_number] -supplier supplier_node
```

The arguments are:

Argument	Description
-connect *net_service_name*	Connects to the database using the net service name defined in the `tnsnames.ora` file
-start *change_number*	Specifies the start change number for the retry operation. If you skip this option, then the command moves all the changes with change numbers less than or equal to the specified end change number back to the retry queue.
-end *change_number*	Specifies the end change number for the retry operation. If you skip this option, then the command moves all the changes with change numbers greater than or equal to the specified start change number back to the retry queue.
-equal *change_number*	Specifies the change number. The command moves the exact change conflict back to the retry queue. This option should not be present when `-start` or `-end` is used.
-supplier *supplier_node*	Specifies the supplier node where the changes originate

Moving a Change from the Human Intervention Queue into the Purge Queue

To place a change into the purge queue, use this syntax:

```
hiqpurge.sh –connect net_service_name [-start change_number] [-end change_
number] [-equal change_number] -supplier supplier_node
```

Arguments are:

Argument	Description
-connect *net_service_name*	Connects to the database using the net service name defined in the tnsnames.ora file
-start *change_number*	Specifies the start change number for the purge operation. If you skip this option, then the command moves all the changes with change numbers less or equal to the specified end change number back to the purge queue.

Argument	Description
`-end` *change_number*	Specifies the end change number for the purge operation. If you skip this option, then the command moves all the changes with change numbers greater or equal to the specified start change number back to the purge queue.
`-equal` *change_number*	Specifies the change number of the change. The command moves the exact change conflict back to the purge queue. This option should not be present when `-start` or `-end` is used.
`-supplier` *supplier_node*	Specifies the supplier node where the changes originate

> **Note:** When using hiqretry.sh or hiqpurge.sh, if you do not want all changes to be moved, then you must supply either the `-equal` flag, or a combination of the `-start` and `-end` flags.

Examples: Using the Human Intervention Queue Manipulation Tool

The following examples illustrate how to use the Human Intervention Queue Manipulation Tool.

Example: Retrying and Discarding Changes

Suppose that, after analyzing the replication log, you decide to do the following:

- Retry changes coming from the supplier node, ldap_rep1, with change numbers between 10324 to 10579
- Discard changes with change numbers between 10581 to 10623.

To do this, you issue these two commands:

```
hiqretry.sh –connect oiddb1 -start 10324 -end 10579 –supplier ldap_rep1
hiqpurge.sh –connect oiddb1 -start 10581 -end 10623 –supplier ldap_repl
```

The first command moves changes originating in ldap_rep1 with change numbers from 10324 to 10579 back to the retry queue. The second command deletes changes that originate in the supplier ldap_repl and that have change numbers from 10581 to 10623.

Example: Moving a Single Change from the Human Intervention Queue to the Retry Queue

The following command moves the change with change number equal to 10519 back to the retry queue.

```
hiqretry.sh –connect oiddb1 –equal 10519 –supplier ldap_repl
```

Example: Moving a Group of Changes from the Human Intervention Queue to the Retry Queue

The following command moves all the changes with change number greater or equal to 10324 back to the retry queue.

```
hiqretry.sh –connect oiddb1 –start 10324 –supplier ldap_repl
```

The following command moves all the changes with change numbers less than or equal to 10579 back to the retry queue.

```
hiqretry.sh –connect oiddb1 –end 10579 –supplier ldap_repl
```

Example: Moving All Changes from the Human Intervention Queue to the Retry Queue

The following command includes no options. It moves all changes that originate in the supplier ldap_repl from the human intervention queue to the retry queue.

```
hiqretry.sh –connect oiddb1 –supplier ldap_repl
```

OID Reconciliation Tool Syntax

When the Oracle directory replication server encounters inconsistent data, you can use the OID Reconciliation Tool to synchronize the entries on the consumer with those on the supplier. When you do this, perform the following general steps:

1. Set the supplier and the consumer to read-only mode.
2. Ensure that the supplier and the consumer are in tranquil state. If they are not in a tranquil state, then wait until they have finished updating.
3. Identify the inconsistent entries or subtree on the consumer.
4. Use the OID Reconciliation Tool to fix the inconsistent entries or subtree on the consumer.
5. Set the participating supplier and consumer back to read-write mode.

Reconciling Inconsistent Data by Using the OID Reconciliation Tool

The OID Reconciliation Tool uses this syntax:

```
oidreconcile -h supplier_host -c consumer_host [-P supplier_port] [-p consumer_
port] [-s scope] -b "basedn" -W supplier_password -w consumer_password [-T
thread]
```

Argument	Description
-h *supplier_host*	Supplier host. This can be a computer name or IP address.
-c *consumer_host*	Consumer host. This can be a computer name or IP address.
-P *supplier_port*	Supplier TCP port. If you do not specify this option, then the tool connects to the default port (389).
-p *consumer_port*	Consumer TCP port. If you do not specify this option, then the tool connects to the default port (389).
-s *scope*	Reconcile scope: subtree
-b "*basedn*"	Specifies the distinguished name of the entry on which to perform reconciliation.
-W *supplier_password*	The password of `cn=orcladmin` of the supplier node
-w *consumer_password*	The password of `cn=orcladmin` of the consumer node
-T *thread*	Worker thread

How the OID Reconciliation Tool Works

When the OID Reconciliation Tool receives the specified DN, it compares the `orclGuid` of the parent DN on both the supplier and the consumer.

If the global identification (`orclGuid`) of both parents match, and the option `-s` *`subtree`* is set, then the OID Reconciliation Tool does the following:

1. Deletes all the entries in the subtree on the consumer node
2. Replaces them with entries from the supplier node

For example, the following command replaces the whole subtree starting from `"ou=hr,o=acme,c=us"` on the consumer with the equivalent subtree on the supplier:

```
oidreconcile -h supplier_host -P 389 -c consumer_host -p 389
-b "ou=hr,o=acme,c=us" -s subtree -W supplier_password -w consumer_password
```

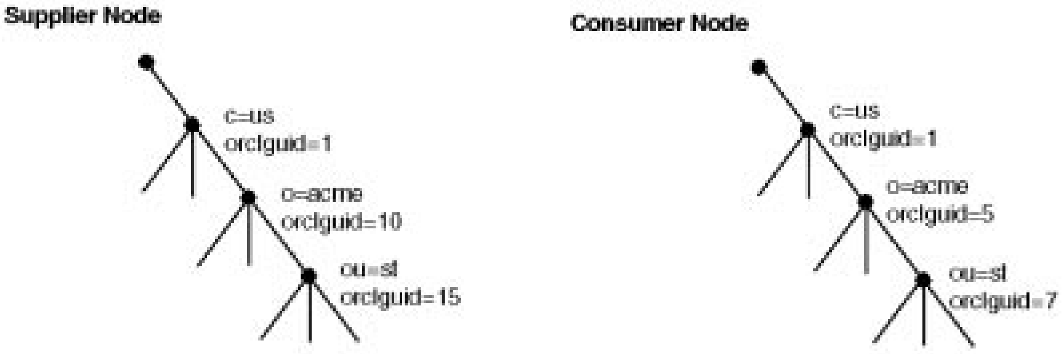
If the global identification (orclGuid) of both parents ("o=acme,c=us") match, and -s subtree is not set, then the OID Reconciliation Tool replaces only the entry itself on the consumer node with the specified entry from the supplier node.

For example, the following command, in which the option "-s subtree" is not set, replaces only the specified entry, "ou=hr,o=acme,c=us".

```
oidreconcile -h supplier -P 389 -c consumer -p 389 -b "ou=hr, o=acme, c=us"
-W supplier_password -w consumer_password
```

The next figure helps to explain how this process works.

Figure 27–1 Example: OID Reconciliation Tool Process



This figure shows two DITs, one on a supplier node and one on a consumer node. In the DIT on the supplier node, the orclGuid for c=us is 1 (one), the orclGuid for o=acme is 10, and the orclGuid for ou=st is 15. On the consumer node, the orclGuid for o=acme is 5, and the orclGuid for ou=st is 7.

The orclGuids for the parent of o=acme,c=us—namely, c=us—on both the supplier and the consumer match. Therefore, the following command replaces all entries under o=acme,c=us on the consumer with the corresponding ones on supplier:

```
oidreconcile -h supplier -c consumer -b "o=acme, c=us" -s subtree -W supplier_
password -w consumer_password
```

If the orclGuid of both parents does not match, then the OID Reconciliation Tool does not perform the reconciliation. Instead, it tells the user the first ancestor on the consumer in which the orclGuid matches that of the same ancestor on the supplier.

For example, in the previous example, suppose you were to run the following command:

```
oidreconcile -h supplier -c consumer -b "ou=st, o=acme, c=us" -s subtree
-W supplier_password -w consumer_password
```

This command would result in a message that the first ancestor of `ou=st` in which the match of the `orclGuid` is `o=acme,c=us`. This message means that you should use `o=acme,c=us` as `basedn` argument for oidreconcile.

OID Database Statistics Collection Tool Syntax

The $*ORACLE_HOME*/ldap/admin/oidstats.sh tool is provided to analyze the various database `ods` schema objects to estimate the statistics.

The OID Database Statistics Collection Tool uses this syntax:

```
oidstats.sh [ -connect net_service_name ]
            [ -login database_account_login ]
            [ -pass database_account_password ]
            [ -all ]
            [ -cat catalog_name ]
            [ -pct percent ]
           [ -help | -usage ]
```

The parameters are:

Parameter	Description	Default
`connect` *`net_service_name`*	DB connect string	*ORACLE_SID*
`login` *`database_account_login`*	Database user name	`ods`
`pass` *`database_account_password`*	Database account password	`ods`
`all`	Estimate statistics on all catalog tables plus DN catalogue	All catalogs
`cat` *`catalog_name`*	Estimate statistics either on all catalogs (all) or on a particular one, for example, ct_cn	None
`pct` *`percent`*	Percent of data to sample	100

Examples: Using the OID Database Statistics Collection Tool

Each of the following examples assume that the ORACLE_*SID* and the default user name and password are in effect.

The following example estimates statistics based on 100 percent sample data of all tables:

```
oidstats.sh -all -pct 100
```

The following example estimates statistics based on 50 percent sample data of all tables:

```
oidstats.sh -all -pct 50
```

The following example estimates statistics based on 50 percent sample data of CT_CN table:

```
oidstats.sh -cat ct_cn -pct 50
```

The following example estimates statistics based on 40 percent sample data of all catalog tables:

```
oidstats.sh -cat all -pct 40
```

B

Using Access Control Directive Format

This appendix describes the format (syntax) of any **access control item (ACI)**. It contains these topics:

- Schema for orclACI
- Schema for orclEntryLevelACI

Schema for orclACI

The access control directive defined by the user attribute `orclACI` has the following schema:

```
OrclACI:
{ object_identifier NAME 'orclACI' DESC 'Stores an inheritable ACI' EQUALITY
accessDirectiveMatch SYNTAX 'accessDirectiveDescription'  USAGE
'directoryOperation'}
```

accessDirectiveDescription has the following BNF:

```
<accessDirectiveDescription>
                  ::= access to <object> [by <subject> ( <accessList> )]+

<object> ::= [attr <EQ-OR-NEQ> (<attrList>) | entry] [filter=(<ldapFilter>)]

<subject> ::= <entity> [<BindMode>]

<entity> ::= * | self | dn="<regex>" | dnAttr=(<dn_attribute>) | group="<dn>"

<BindMode> ::= | BindMode = Simple
               | BindMode = SSLNoauth
               | BindMode = SSLOneway
               | BindMode = SSLTwoway

<accessList> ::= <access> | <access>, <accessList>

<access> ::= none | compare | search | browse | read | selfwrite | write | add |
delete | nocompare | nosearch | nobrowse |noread | noselfwrite | nowrite | noadd
| nodelete

<attrList> ::=  * | <attribute name> | <attribute name>,<attrList>

<EQ-OR-NEQ> ::=  = | !=

<regex> ::= <dn> | *,<dn_of_any_subtree_root>
```

> **Note:** The regular expression defined above is not meant to match any arbitrary expression. The syntax only allows expressions where the wild card is followed by a comma and a valid DN. The latter DN denoted by <*dn_of_any_subtree_root*> is intended to specify the root of some subtree.

Schema for orclEntryLevelACI

The entry level access control directive defined by the user attribute `orclEntryLevelACI` has the following schema:

```
"orclEntryLevelACI":
{ object_identifier NAME 'orclEntryLevelACI' DESC 'Stores entry level ACL
Directive'
EQUALITY accessDirectiveMatch SYNTAX 'orclEntryLevelACIDescription'
USAGE 'directoryOperation' }


<orclEntryLevelACIDescription>
::= access to <object> [by <subject> ( <accessList> )]+
```

C

Schema Elements

This appendix briefly lists different schema elements supported by Oracle Internet Directory. Most of these elements are used as defined by the ldapext and ASID working groups of the Internet Engineering Task Force (IETF).

See Also: The following URLs on the World Wide Web:

- `http://www.ietf.org` for the IETF home page
- `http://www.ietf.org/html.charters/ldapext-charter.html` for the ldapext charter and LDAP drafts)
- `http://www.ietf.org/html.charters/ldup-charter.html` for the LDUP charter and drafts
- `http://www.iana.org`, the Internet Assigned Numbers Authority home page, for information about object identifiers

This appendix contains these topics:

- IETF Requests for Comments (RFCs) Enforced by Oracle Internet Directory
- IETF Drafts Enforced by Oracle Internet Directory
- Proprietary Oracle Internet Directory Schema Elements
- LDAP Syntax
- Matching Rules

IETF Requests for Comments (RFCs) Enforced by Oracle Internet Directory

Oracle Internet Directory enforces the following Requests for Comments (RFCs) of the Internet Engineering Task Force (IETF):

RFC	Title	URL
1777	Lightweight Directory Access Protocol	http://www.ietf.org/rfc/rfc1777.txt
1778	The String Representation of Standard Attribute Syntaxes	http://www.ietf.org/rfc/rfc1778.txt
1779	A String Representation of Distinguished Names	http://www.ietf.org/rfc/rfc1779.txt
1960	A String Representation of LDAP Search Filters	http://www.ietf.org/rfc/rfc1960
2079	Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs)	http://www.ietf.org/rfc/rfc2079.txt
2247	Using Domains in LDAP/X.500 Distinguished Names	http://www.ietf.org/rfc/rfc2247.txt
2251	Lightweight Directory Access Protocol (v3)	http://www.ietf.org/rfc/rfc2251.txt
2252	Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions	http://www.ietf.org/rfc/rfc2252.txt
2253	Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names	http://www.ietf.org/rfc/rfc2253.txt
2254	The String Representation of LDAP Search Filters	http://www.ietf.org/rfc/rfc2254.txt
2255	The LDAP URL Format	http://www.ietf.org/rfc/rfc2255.txt
2256	A Summary of the X.500(96) User Schema for use with LDAPv3	http://www.ietf.org/rfc/rfc2256.txt

IETF Drafts Enforced by Oracle Internet Directory

Oracle Internet Directory enforces the following two drafts of the IETF:

Draft:	"Definition of the inetOrgPerson LDAP Object Class"
URL:	http://ietf.org/rfc/rfc2798.txt

Draft :	"Referrals and Knowledge References in LDAP Directories"
URL:	http://www.ietf.org/proceedings/99nov/I-D/draft-ietf-ldapext-knowledge-00.txt

Proprietary Oracle Internet Directory Schema Elements

Oracle Internet Directory's proprietary schema includes attributes and object classes in these categories:

- Access Control
- Replication
- Oracle Internet Directory Configuration
- SSL
- Audit Log
- Configuration Set Entry Attributes

In addition, Oracle Internet Directory installation includes schema elements that enable specific Oracle products to use Oracle Internet Directory. For information about these schema elements, see the documentation for the specific Oracle product.

Access Control

Attributes	`orclEntryLevelACI, orclACI`
Object Class	`orclPrivilegeGroup`

Replication

Attributes orclGUID, changeNumber changeType, changes, orclParentGUID, server, supplier, consumer, orclReplBindDN, orclReplBindPassword, changeLog, changeStatus, orclChangeRetryCount, orclPurgeSchedule, orclDirReplGroupAgreement, orclAgreementId, orclSupplierReference,orclConsumerReference, orclReplicationProtocol, orclUpdateSchedule, targetDN, orclExcludedNamingcontexts, orclDirReplGroupDSAs

Object class changeLogEntry, changeStatusEntry, orclReplAgreementEntry

Oracle Internet Directory Configuration

Attributes orcldebugflag, orclMaxCC, orclDBType, orclSuffix, orclDITRoot, orclSuName, orclSuPassword, orclSizeLimit, orclTimeLimit, orclGuName, orclGuPassword, orclServerProcs, orclconfigsetnumber, orclhostname, orclIndexedAttribute, orclCatalogEntryDN, orclServerMode, orclPrName, orclPrPassword, orclUseEncrypt, orclDirectoryVersion

Object class subconfig, orclConfigSet, orclLDAPSubConfig, orclREPLSubConfig, orclcontainerOC, subregistry, orclLDAPInstance, orclREPLInstance, orclIndexOC, orcleventLog, orclEvents

SSL

Note: These attribute values are stored as part of configuration entries.

Attributes orclsslAuthentication, orclsslEnable, 'orclsslWalletURL, orclsslWalletPasswd, orclsslPort, orclsslVersion

Audit Log

Attributes	`orclServerEvent, orcleventtype, orclauditattribute, orclauditmessage, orcleventtime, orcluserdn, orclSequence, orclAuditLevel, orclOpResult`
Object class	`OrclAuditOC`

Configuration Set Entry Attributes

The following table lists and describes the entire set of configuration set entry attributes that are used to configure an instance of a directory server.

Parameter	Description
`orcldebugflag`	Debug level associated with this instance of the server. The default for configset0 is 0. The range is 0 to 65535.
`orclmaxcc`	Maximum number of concurrent database connections. The default for configset0 is 10. You cannot use a negative value for this attribute.
`orclserverprocs`	Number of server processes to start. The default for configset0 is 1. You cannot use a negative value for this attribute.
`orclsslport`	SSL mode default port (default 636). When you run the directory in the secure mode, it listens at default port 636 and accepts only SSL-based TCP/IP connections. (When you run the directory in the normal mode, it listens at default port 389, accepting normal TCP/IP connections.) You might want to change this port when you add multiple LDAP server instances.
`orclnonsslport`	Non-SSL mode default port (default 389).
`orclsslenable`	Flag for toggling SSL on and off. You would want to toggle this flag when you use different instances of the same server for either SSL or non-SSL. You may use either of the following two values: ■ 0 = disables SSL (default in configuration set0) ■ 1 = enables SSL The default is 0.

Parameter	Description
`orclsslauthentication`	Flag, with values of 1, 32, or 64, for specifying the type of authentication you elect to use for each instance of the Oracle directory server. The default value, 1, specifies no authentication. You can run different values concurrently for different instances. Values of one-way and two-way authentication require wallets. You may use one of the following three values: ■ 1 = no SSL authentication ■ 32 = one-way SSL authentication (the server sends its certificate to the client) ■ 64 = two-way SSL authentication (client and server send certificates to each other)

`orclsslwalleturl`	Sets the location of the Oracle wallet. You initially set this value when you create the wallet. If you elect to change the location of the Oracle wallet, you must change this parameter. You must set the wallet location on both the client and the server. For example, on Solaris, you could set this parameter as follows:
	`orclsslwalleturl=file:/Home/my_dir/`
	On Windows NT, you could set this parameter as follows:
	`file:Home\my_dir\`

`orclsslwalletpasswd`	Password used by the server to open its wallet. You initially set this value when you create the wallet. If you elect to change the wallet password, you must change this parameter. You must set the wallet password on both the client and the server.
`orclsslversion`	SSL version. The default is 3.

See Also:

- "Setting Debug Logging Levels by Using the OID Control Utility" on page 6-27 for information on debug levels
- Appendix D, "Using Oracle Wallet Manager" for information on setting the location of the Oracle Wallet and the Oracle Wallet password

LDAP Syntax

Syntax defines the type of values that an attribute can hold. Oracle Internet Directory recognizes most of the syntax specified in RFC 2252, that is, it allows you to associate most of the syntax described in that document with an attribute. In addition to recognizing most LDAP syntax, Oracle Internet Directory enforces some LDAP syntax.

This section covers topics in the following subsections:

- LDAP Syntax Enforced by Oracle Internet Directory
- Commonly Used LDAP Syntax Recognized by Oracle Internet Directory
- Additional LDAP Syntax Recognized by Oracle Internet Directory
- Size of Attribute Values

LDAP Syntax Enforced by Oracle Internet Directory

Oracle Internet Directory enforces LDAP syntax for the following:

- DN
- Facsimile Telephone Number
- OID (object identifier)
- Telephone Number

Note: The values you specify for these attributes must conform to the syntax specified in RFC 2252.

Commonly Used LDAP Syntax Recognized by Oracle Internet Directory

The following LDAP syntax is more commonly used:

Attribute Type Description

Boolean

Certificate

Directory String

DN

Facsimile Telephone Number

INTEGER

JPEG

Name And Optional UID

Numeric String

Object Class Description

Octet String

OID

Presentation Address

Printable String

Telephone Number

UTC Time

Additional LDAP Syntax Recognized by Oracle Internet Directory

In addition to the commonly used LDAP syntax defined above, Oracle Internet Directory recognizes LDAP syntax for the following:

Access Point

ACI Item

Audio

Binary

Bit String

Certificate List

Certificate Pair

Country String

Data Quality Syntax

Delivery Method

DIT Content Rule Description

DIT Structure Rule Description

DL Submit Permission

DSA Quality Syntax

DSE Type

Enhanced Guide

Fax

Generalized Time

Guide

IA5 String

LDAP Schema Definition

LDAP Schema Description

LDAP Syntax Description

Mail Preference

Master And Shadow Access Points

Matching Rule

Matching Rule Use Description

MHS OR Address

Modify Rights

Name Form Description

Object Class Description

Octet String

Other Mailbox

Postal Address

Protocol Information

Substring Assertion

Subtree Specification

Supplier And Consumer

Supplier Information

Supplier Or Consumer

Supported Algorithm

Teletex TerminalIdentifier

Telex Number

Size of Attribute Values

Syntax does not put any specific size constraint on attribute values. You can, however, use syntax to specify the size of the attribute value. Oracle Internet Directory does not enforce the 'len' characteristics on the attribute.

For example, to limit an attribute foo to a size of 64, you would define the attribute as follows:

```
(object_identifier_of_attribute NAME 'foo' EQUALITY caseIgnoreMatch SYNTAX
'object_identifier_of_syntax{64}')
```

See Also: Section 4.1.6 f of RFC2251 for more information on Attribute Value. You can find this RFC at the following URL: http://www.ietf.org/rfc/rfc2251.txt.

Matching Rules

Oracle Internet Directory recognizes the following matching rules definitions in the schema.

accessDirectiveMatch

bitStringMatch

caseExactMatch

caseExactIA5Match

caseIgnoreIA5Match

caseIgnoreListMatch

caseIgnoreMatch

caseIgnoreOrderingMatch

distinguishedNameMatch

generalizedTimeMatch

generalizedTimeOrderingMatch

IntegerMatch

numericStringMatch

objectIdentifierFirstComponentMatch

ObjectIdentifierMatch

OctetStringMatch

presentationAddressMatch

protocolInformationMatch

telephoneNumberMatch

uniqueMemberMatch

Of the matching rules in the previous list, Oracle Internet Directory actually enforces the following when it compares attribute values:

distinguishedNameMatch

caseExactMatch

caseIgnoreMatch

numericStringMatch

IntegerMatch

telephoneNumberMatch

D

Using Oracle Wallet Manager

Security administrators use Oracle Wallet Manager to manage public-key security credentials on Oracle clients and servers. The wallets it creates are opened by using either the Oracle Enterprise Login Assistant or the Oracle Wallet Manager.

This chapter describes the Oracle Wallet Manager, in the following sections:

> **See Also:** *Oracle Advanced Security Administrator's Guide* for information about how to open and close wallets for secure SSL communications by using Oracle Enterprise Login Assistant

Overview

Traditional private-key or symmetric-key cryptography requires that entities desiring to establish secure communications possess a single secret key known only to them. *Harriet* and *Dick*, for example, could agree to shift each letter in their private messages by two character positions (A becomes C, B becomes E, and so on) to encrypt the message text. Using this method, a *HELLO* message from Harriet to Dick would read *JGNNP*. The actual encryption methods in current use are much more complex and significantly more secure, but an underlying problem remains—sending messages encrypted with a single key requires prior, *secure* distribution of the key to each participating party. Otherwise, a malicious third party might obtain the key, intercept communications, and compromise security. Public-key cryptography addresses this problem, by providing a secure method for key distribution.

Public-key cryptography requires a party to possess a **public/private key pair**. The **private key** is kept secret and is known only to that party. The **public key**, as the name implies, is freely available. To send a secret message to this party requires that a third party sender encrypt the message with the public key. Such a message can only be decrypted by a party holding the associated private key.

For example, when Dick wants to send a secure message to Harriet, he first asks Harriet for her public key (or obtains it from another, public source). Harriet gives Dick the public key, but Tom, a malicious eavesdropper, also obtains the public key. Nevertheless, when Dick sends Harriet a message encrypted with her public key, Tom cannot decrypt it; the message can only be decrypted with Harriet's private key.

Public-key algorithms thus guarantee the secrecy of a message, but they don't guarantee *secure communications* because they don't verify the identities of the communicating parties. In order to establish secure communications, it is important to verify that the public key used to encrypt a message does in fact belong to the target recipient. Otherwise, a third party can potentially eavesdrop on the communication and intercept public key requests, substituting its public key for a legitimate key.

If Tom, for example, is able to substitute his public key for Harriet's public key and send it to Dick, Dick might then send a message to Harriet encrypted with Tom's public key—believing he was using Harriet's public key. Tom could then decrypt a subsequent intercepted message from Dick using his private key, re-encrypt it with Harriet's public key and re-transmit it to Harriet. Harriet could then decrypt the incoming message using her private key, and never know that it had been intercepted by Tom.

In order to avoid such a man-in-the-middle attack, it is necessary to verify the owner of the public key, a process called **authentication**. This authentication can be accomplished through a **certificate authority (CA)**.

A CA is a third party that is trusted by both of the parties attempting secure communication. The CA issues public key certificates that contain an entity's name, public key, and certain other security credentials. Such credentials typically include the CA name, the CA signature, and the certificate effective dates (From Date, To Date).

The CA uses its private key to encrypt a message, while the public key is used to decrypt it, thus verifying that the message was encrypted by the CA. The CA public key is well known, and does not have to be authenticated each time it is accessed. Such CA public keys are stored in a **wallet**.

Oracle Wallet Manager is a stand-alone Java application that wallet owners use to manage and edit the security credentials in their Oracle wallets. These tasks include the following:

- Generating a public/private key pair and creating a certificate request for submission to a CA.
- Installing a certificate for the entity.
- Configuring **trusted certificates** for the entity.
- Opening a wallet to enable access to PKI-based services.
- Creating a wallet that can be accessed by using either Oracle Enterprise Login Assistant or Oracle Wallet Manager.

Managing Wallets

This section describes how to create a new wallet and perform associated wallet management tasks, such as generating certificate requests, exporting certificate requests, and importing certificates into wallets, in the following subsections:

- Starting Oracle Wallet Manager
- Creating a New Wallet
- Opening an Existing Wallet
- Closing a Wallet
- Saving Changes
- Saving the Open Wallet to a New Location
- Saving in System Default
- Deleting the Wallet
- Changing the Password
- Using Auto Login
- Using Oracle Wallet Manager with Oracle Application Server

Starting Oracle Wallet Manager

To start Oracle Wallet Manager:

UNIX:	Enter `owm` at the command line.
Windows NT:	Press Start > *ORACLE_HOME* > Network Administration > Wallet Manager

Creating a New Wallet

Create a new wallet as follows:

1. Choose `Wallet > New` from the menu bar; the New Wallet dialog box appears.
2. Read the recommended guidelines for creating a password and enter a password in the Wallet Password field.

   Because an Oracle wallet contains a user's credentials that can be used to authenticate the user to multiple databases, it is especially important to choose a strong password for the wallet. A malicious user who guesses the password to a user's wallet can access all the databases that the user can access.

Oracle Corporation recommends that you choose a password that is not too short, not easily guessed, and is reasonably complex. A reasonably complex password has at least six characters, and contains at least one symbol or number—so that it will not be found in a dictionary.

Example: `gol8fer`

It is also a prudent security practice for users to change their passwords periodically, such as once a month, or once a quarter.

3. Re-enter that password in the Confirm Password field.
4. Choose `OK` to continue.
5. An Alert is displayed, and informs you that a new empty wallet has been created. It prompts you to decide whether you want to create a certificate request. See: "Creating a Certificate Request" on page D-9.

   If you choose Cancel, you are returned to the Oracle Wallet Manager main window. The new wallet you just created appears in the left window pane. The certificate has a status of `Empty`, and the wallet displays its default trusted certificates.
6. Select `Wallet > Save In System Default` to save the new wallet.

   If you do not have permission to save the wallet in the system default, you can save it to another location.

   A message at the bottom of the window informs you that the wallet was successfully saved.

Opening an Existing Wallet

Open a wallet that already exists in the file system directory as follows:

1. Choose `Wallet > Open` from the menu bar; the Select Directory dialog box appears.
2. Navigate to the directory location in which the wallet is located, and select the directory.
3. Choose OK; the Open Wallet dialog box appears.
4. Enter the wallet password in the Wallet Password field.

5. Choose `OK`.
6. The message `Wallet opened successfully` appears at the bottom of the window, and you are returned to the Oracle Wallet Manager main window. The wallet's certificate and its trusted certificates are displayed in the left window pane.

Closing a Wallet

To close an open wallet in the currently selected directory:

- Choose `Wallet > Close`.
- The message `Wallet closed successfully` appears at the bottom of the window, to confirm that the wallet is closed.

Saving Changes

To save your changes to the current open wallet:

- Choose `Wallet > Save`.
- A message at the bottom of the window confirms that the wallet changes were successfully saved to the wallet in the selected directory location.

Saving the Open Wallet to a New Location

Use the `Save As` option to save the current open wallet to a new directory location:

1. Choose `Wallet > Save As`. The select directory dialog box appears.
2. Select a directory location to save the wallet.
3. Choose `OK`.

   The following message appears if a wallet already exists in the selected directory:

   ```
   A wallet already exists in the selected path. Do you
   want to overwrite it?.
   ```

   Choose `Yes` to overwrite the existing wallet, or `No` to save the wallet to another directory.

   A message at the bottom of the window confirms that the wallet was successfully saved to the selected directory location.

Saving in System Default

Use the `Save in System Default` menu option to save the current open wallet to the system default directory location. This makes the current open wallet the wallet that is used by SSL:

- Choose `Wallet > Save in System Default`.
- A message at the bottom of the window confirms that the wallet was successfully saved in the system default wallet location.

Deleting the Wallet

To delete the current open wallet:

1. Choose `Wallet > Delete`; the `Delete Wallet` dialog box appears.
2. Review the displayed wallet location to verify you are deleting the correct wallet.
3. Enter the wallet password.
4. Choose `OK`; a dialog panel appears to inform you that the wallet was successfully deleted.

Note: Any open wallet in application memory will remain in memory until the application exits. Therefore, deleting a wallet that is currently in use does not immediately affect system operation.

Changing the Password

A password change is effective immediately. The wallet is saved to the currently selected directory, with the new encrypted password.To change the password for the current open wallet:

1. Choose `Wallet > Change Password;` the `Change Wallet Password` dialog box appears.
2. Enter the existing wallet password.
3. Enter the new password.
4. Re-enter the new password.
5. Choose `OK`.

A message at the bottom of the window confirms that the password was successfully changed.

Using Auto Login

The Oracle Wallet Manager Auto Login feature opens a copy of the wallet and enables PKI-based access to secure services—as long as the wallet in the specified directory remains open in memory.

You must enable Auto Login if you want single sign-on access to multiple Oracle databases.

Enabling Auto Login

To enable Auto Login:

1. Choose `Wallet` from the menu bar.
2. Choose the check box next to the Auto Login menu item; a message at the bottom of the window displays `Autologin enabled`.

Disabling Auto Login

To disable Auto Login:

1. Choose `Wallet` from the menu bar.
2. Choose the check box next to the Auto Login menu item; a message at the bottom of the window displays `Autologin disabled`.

Using Oracle Wallet Manager with Oracle Application Server

When using the Oracle Application Server (OAS), you must install the Oracle Wallet Manager on a primary node and on each remote node in a multi-node configuration. After you install the product on each node you must then copy the wallet from the primary node to each of the remote nodes.

Managing Certificates

Oracle Wallet Manager uses two kinds of certificates: user certificates and trusted certificates. This section describes how to manage both certificate types, in the following subsections:

- Managing User Certificates
- Managing Trusted Certificates

Note: You must first install a trusted certificate from the certificate authority before you can install a user certificate issued by that authority. Several trusted certificates are installed by default when you create a new wallet.

Managing User Certificates

Managing user certificates involves the following tasks:

- Creating a Certificate Request
- Exporting a User Certificate Request
- Importing the User Certificate into the Wallet
- Removing a User Certificate from a Wallet

Creating a Certificate Request

The actual certificate request becomes part of the wallet. You can reuse any certificate request to obtain a new certificate. However, you cannot edit an existing certificate request; store only a correctly filled out certificate request in a wallet.

To create a PKCS #10 certificate request:

1. Choose `Operations > Create Certificate Request`; the `Create Certificate Request` dialog box appears.
2. Enter the following information (Table D–1):

Table D–1 Certificate Request: Fields and Descriptions

Field Name	Description
Common Name	Mandatory. Enter the name of the user's or service's identity. Enter a user's name in first name /last name format.

Table D–1 Certificate Request: Fields and Descriptions

Field Name	Description
Organizational Unit	Optional. Enter the name of the identity's organizational unit. Example: Finance.
Organization	Optional.Enter the name of the identity's organization. Example: XYZ Corp.
Locality / City	Optional. Enter the name of the locality or city in which the identity resides.
State / Province	Optional. Enter the full name of the state or province in which the identity resides. Enter the full state name, because some certificate authorities do not accept two–letter abbreviations.
Country	Mandatory. Choose the drop-down list to view a list of country abbreviations. Select the country in which the organization is located.
Key Size	Mandatory. Choose the drop-down box to view a list of key sizes to use when creating the public / private key pair.
Advanced	Optional. Choose `Advanced` to view the Advanced Certificate Request dialog panel. Use this field to edit or customize the identity's distinguished name (DN). For example, you can edit the full state name and locality.

3. Choose `OK`. An Oracle Wallet Manager dialog box informs you that a certificate request was successfully created. You can either copy the certificate request text from the body of this dialog panel and paste it into an e-mail message to send to a certificate authority, or you can export the certificate request to a file.
4. Choose `OK`. You are returned to the Oracle Wallet Manager main window; the status of the certificate is changed to `Requested`.

Exporting a User Certificate Request

Save the certificate request in a file system directory when you elect to export a certificate request:

1. Choose `Operations > Export Certificate Request` from the menu bar; the Export Certificate Request dialog box appears.
2. Enter the file system directory in which you want o save your certificate request, or navigate to the directory structure under Folders.
3. Enter a file name to save your certificate request, in the Enter File Name field.
4. Choose `OK`. A message at the bottom of the window confirms that the certificate request was successfully exported to the file. You are returned to the Oracle Wallet Manager main window.

Importing the User Certificate into the Wallet

You will receive an e-mail notification from the certificate authority informing you that your certificate request has been fulfilled. Import the certificate into a wallet in either of two ways: copy and paste the certificate from the e-mail you receive from the certificate authority, or import the user certificate from a file.

Pasting the Certificate

To paste the certificate:

1. Copy the certificate text from the e-mail message or file you receive from the certificate authority. Include the lines `Begin Certificate` and `End Certificate.`
2. Choose `Operations > Import User Certificate` from the menu bar; the Import Certificate dialog box appears.
3. Choose the `Paste the Certificate` button, and choose `OK`; an Import Certificate dialog box appears with the following message:

   ```
   Please provide a base64 format certificate and paste it below.
   ```
4. Paste the certificate into the dialog box, and choose `OK`. A message at the bottom of the window confirms that the certificate was successfully installed. You are returned to the Oracle Wallet Manager main panel, and the wallet status changes to `Ready`.

Selecting a File that Contains the Certificate

To select the file:

1. Choose `Operations > Import User Certificate` from the menu bar.
2. Choose the `Select a file...` certificate button, and choose `OK`; the Import Certificate dialog box appears.
3. Enter the path or folder name of the certificate location.
4. Select the name of the certificate file (for example, `cert.txt`).
5. Choose `OK`. A message at the bottom of the window appears, to inform you that the certificate was successfully installed. You are returned to the Oracle Wallet Manager main panel, and the wallet status is changes to `Ready`.

Removing a User Certificate from a Wallet

1. Choose `Operations > Remove User Certificate`; a dialog panel appears and prompts you to verify that you want to remove the user certificate from the wallet.
2. Choose `Yes`; you are returned to the Oracle Wallet Manager main panel, and the certificate displays a status of `Requested`.

Managing Trusted Certificates

Managing trusted certificates includes the following tasks:

- Importing a Trusted Certificate
- Removing a Trusted Certificate
- Exporting a Trusted Certificate
- Exporting All Trusted Certificates
- Exporting a Wallet

Importing a Trusted Certificate

You can import a trusted certificate into a wallet in either of two ways: paste the trusted certificate from an e-mail that you receive from the certificate authority, or import the trusted certificate from a file.

Oracle Wallet Manager automatically installs trusted certificates from VeriSign, RSA, and GTE CyberTrust Entrust when you create a new wallet.

Pasting the Trusted Certificate To paste the trusted certificate:

1. Choose `Operations > Import Trusted Certificate` from the menu bar; the Import Trusted Certificate dialog panel appears.
2. Choose the `Paste the Certificate` button, and choose `OK`. An Import Trusted Certificate dialog panel appears with the following message:

   ```
   Please provide a base64 format certificate and paste it below.
   ```
3. Copy the trusted certificate from the body of the e-mail message you received that contained the user certificate. Include the lines `Begin Certificate` and `End Certificate.`
4. Paste the certificate into the window, and Choose `OK`. A message at the bottom of the window informs you that the trusted certificate was successfully installed.
5. Choose `OK`; you are returned to the Oracle Wallet Manager main panel, and the trusted certificate appears at the bottom of the Trusted Certificates tree.

Selecting a File that Contains the Trusted Certificate

To select the file:

1. Choose `Operations > Import Trusted Certificate` from the menu bar. The Import Trusted Certificate dialog panel appears.
2. Enter the path or folder name of the trusted certificate location.
3. Select the name of the trusted certificate file (for example, `cert.txt`).
4. Choose `OK`. A message at the bottom of the window informs you that the trusted certificate was successfully imported into the wallet.
5. Choose `OK` to exit the dialog panel; you are returned to the Oracle Wallet Manager main panel, and the trusted certificate appears at the bottom of the Trusted Certificates tree.

Removing a Trusted Certificate

To remove a trusted certificate from a wallet:

1. Select the trusted certificate listed in the Trusted Certificates tree.
2. Choose `Operations > Remove Trusted Certificate` from the menu bar.

A dialog panel warns you that your user certificate will no longer be verifiable by its recipients if you remove the trusted certificate that was used to sign it.

3. Choose `Yes`; the selected trusted certificate is removed from the Trusted Certificates tree.

Note: A certificate that is signed by a trusted certificate is no longer verifiable when you remove it from your wallet.

Also, you cannot remove a trusted certificate if it has been used to sign a user certificate that is still present in the wallet. To remove such a trusted certificate, you must first remove the certificates that it has signed.

Exporting a Trusted Certificate

To export a trusted certificate to another file system location:

1. Select `Operations > Export Trusted Certificate`; the Export Trusted Certificate dialog box appears.
2. Select a file system directory to save your trusted certificate, or choose `Browse` to display the directory structure.
3. Enter a file name to save your trusted certificate.
4. Choose `OK`; you are returned to the Oracle Wallet Manager main window.

Exporting All Trusted Certificates

To export all of your trusted certificates to another file system location:

1. Choose `Operations > Export All Trusted Certificates`. The Export Trusted Certificate dialog box appears.
2. Select the file system directory to save your trusted certificates, or choose `Browse` to display the directory structure.
3. Enter a file name to save your trusted certificates.
4. Choose `OK`; you are returned to the Oracle Wallet Manager main window.

Exporting a Wallet

You can export a wallet to text-based PKI formats. Individual components are formatted according to the following standards (Table D–2):

Table D–2 PKI Wallet Encoding Standards

Component	Encoding Standard
Certificate chains	X509v3
Trusted certificates	X509v3
Private keys	PKCS5

E

Upgrading from Oracle Internet Directory Release 2.1.1

This chapter tells you how to upgrade to Oracle Internet Directory release 3.0.1 from Oracle Internet Directory release 2.1.1.

This appendix contains these topics:

- Tasks Before Upgrading
- Upgrading in a Single Node Environment
- Upgrading in a Multi-Node Environment

You can upgrade one node at a time until you have upgraded all of the nodes in the replication group. In a replicated environment, a node running release 3.0.1 can co-exist with nodes running previous releases of Oracle Internet Directory. Moreover, in a replicated environment, upgrade of one node to release 3.0.1 requires no network downtime. The other nodes can remain available while the upgrade progresses.

Tasks Before Upgrading

Before you upgrade, perform the following tasks:

- Task 1: Stop the Oracle Directory Replication Server on the Node to be Upgraded

  See Also: "Stopping an Oracle Directory Replication Server Instance" on page 4-7

- Task 2: Stop the Oracle Directory Server on the Node to be Upgraded

  See Also: "Stopping an Oracle Directory Server Instance" on page 4-5

- Task 3: Stop OID Monitor on the Node to be Upgraded

  See Also: "Stopping the OID Monitor" on page 4-3

Upgrading in a Single Node Environment

To upgrade on a single node, follow the instructions in the installation documentation for your operating system.

Upgrading in a Multi-Node Environment

Upgrading a multi-node Oracle Internet Directory system to release 3.0.1 requires special attention. This section discusses the two ways to upgrade a multi-node Oracle Internet Directory system. It contains these topics:

- Upgrading One Node at a Time
- Upgrading All the Nodes at the Same Time

Upgrading One Node at a Time

Use this method if you do not want any system downtime. While the upgrade on one node is in progress, it allows all the other nodes to remain available. However, using this method requires that you clearly understand and strictly follow these guidelines:

- When you are upgrading a replication network one node at a time, the upgrade is not complete until all nodes are upgraded. However, during this period, all network nodes, except the one being upgraded, remain available.
- While the upgrade is going on, only one node should be Read-Write. The rest should be Read-Only.
- Be sure to perform the upgrade on the **master definition site (MDS)** before you upgrade the master sites.

Perform the following tasks, first on the MDS, then on the master sites.

Task 1: Verify that You Have Stopped the Oracle Internet Directory Processes

See Also: "Tasks Before Upgrading" on page E-2

Task 2: Delete Jobs on Other Nodes

Before shutting down the database at the MDS, run the script delasrjobs.sql located in `$`*`ORACLE_HOME/`*`ldap/admin` on the installation CD. This script deletes **Oracle9i Replication** jobs on other master sites that push changes to the MDS. Deleting these jobs temporarily removes the MDS from the replication environment so that no changes can be applied to it. Other nodes, however, remain operational and continue replicating changes.

Task 3: Shutdown Database and Listener on the Node to be Upgraded

If you do not shutdown the database and listener, then Oracle Universal Installer prompts you to do it.

See Also:

- *Oracle Net Services Administrator’s Guide* for instructions on stopping the listener
- *Oracle9i Database Administrator’s Guide* for instructions on shutting down the database server

Task 4: Upgrade the Node to Oracle Internet Directory Release 3.0.1

Run Oracle Universal Installer to upgrade to Oracle Internet Directory release 3.0.1, which uses Oracle9*i* release 9.0.1. The installer both migrates the database and upgrades Oracle Internet Directory.

Task 5: Verify that the Database and Listener Are Running

After the upgrade is completed, the database and listener are started automatically. Verify that they are running.

See Also:

- *Oracle Net Services Administrator's Guide* for instructions on starting the listener
- *Oracle9i Database Administrator's Guide* for instructions on starting the database server

Test the connectivity to other nodes. If connectivity is broken, then use the backup copies of `listener.ora`, `sqlnet.ora` and `tnsnames.ora` and restart the listener. The backup files are named `listener`*`date`*`.bak`, `sqlnet`*`date`*`.bak` and `tnsnames`*`date`*`.bak`.

Task 6: Create Push Jobs on Other Nodes

After you have upgraded the node, create jobs on other nodes. You do this by executing `$`*`ORACLE_HOME/`*`ldap/admin/creasrjobs.sql` on the upgraded node. This script creates on the other nodes the jobs that were deleted in "Task 2: Delete Jobs on Other Nodes" on page E-3. These jobs now start pushing the existing changes and new changes on other nodes to the node you have just upgraded.

Task 7: Verify that the Oracle Internet Directory Processes Are Running

The upgrade process automatically starts the OID Monitor, the directory server, and the directory replication server. Be sure that these processes are running.

See Also:

- "Starting the OID Monitor" on page 4-2
- "Starting an Oracle Directory Server Instance" on page 4-4
- "Starting an Oracle Directory Replication Server Instance" on page 4-6

Task 8: Upgrade Other Master Sites

After upgrading the MDS, upgrade other master sites one at a time. Perform tasks 1 through 12 on each master site until all the nodes are upgraded.

> **See Also:** Chapter 15, "Managing Directory Replication" for information about the MDS

Upgrading All the Nodes at the Same Time

Use this method to upgrade all the nodes at the same time. If you use this method, then the system is unavailable during the upgrade process.

Task 1: Set All the Nodes in the Network to Read-Only Mode

1. Edit the input file as follows:

   ```
   dn:
   changetype:modify
   replace:orclservermode
   orclservermode:r
   ```

2. Run the following command against all the nodes in the replication network:

   ```
   ldapmodify -D "cn=orcladmin" -w welcome -h host_name -p port_number -f
   input_file.ldif
   ```

Task 2: Wait Until All the Changes in the Change Log Queue Have Been Applied

Before moving to next step, wait for the change log queue to empty. If you skip this step, then changes in the change log queue will be applied once the nodes are upgraded.

Task 3: Verify that You Have Stopped the Oracle Internet Directory Processes

> **See Also:** "Tasks Before Upgrading" on page E-2

Task 4: Shutdown the Database and the Listener on All Nodes

If you do not shutdown the database and listener, then Oracle Universal Installer prompts you to do it.

See Also:

- *Oracle Net Services Administrator's Guide* for instructions on stopping the listener
- *Oracle9i Database Administrator's Guide* for instructions on shutting down the database server

Task 5: Upgrade All the Nodes to Oracle Internet Directory Release 3.0.1

Run Oracle Universal Installer to upgrade to Oracle Internet Directory release 3.0.1, which uses Oracle9*i* release 9.0.1. The installer both migrates the database and upgrades Oracle Internet Directory.

Task 6: Start the Database and Listener on All Nodes

After the upgrade is completed, the database and listener are started automatically. Verify that they are running.

See Also:

- *Oracle Net Services Administrator's Guide* for instructions on starting the listener
- *Oracle9i Database Administrator's Guide* for instructions on starting the database server

Test the connectivity to other nodes. If connectivity is broken, then use the backup copies of `listener.ora`, `sqlnet.ora` and `tnsnames.ora` and restart the listener. The backup files are named `listener`*`date`*`.bak`, `sqlnet`*`date`*`.bak` and `tnsnames`*`date`*`.bak`.

Task 7: Verify that the Oracle Internet Directory Processes Are Running

The upgrade process automatically starts the OID Monitor, the directory server, and the directory replication server. Be sure that these processes are running.

See Also:

- "Starting the OID Monitor" on page 4-2
- "Starting an Oracle Directory Server Instance" on page 4-4
- "Starting an Oracle Directory Replication Server Instance" on page 4-6

LDIF-Based Upgrading

Normally, you do not need to perform LDIF-based upgrading. Use this method when you cannot successfully run the database-based upgrade process.

Oracle Corporation recommends that you use the LDIF-based backup procedure to backup your existing release of Oracle Internet Directory. This is explained in this section.

The LDIF-based upgrade process requires the following procedures on a node being upgraded:

Task 1: Backup the Older Version of Oracle Internet Directory

Be sure that the directory server is not running, then run the script backup_oid.sh located in the `$`*`ORACLE_HOME/`*`ldap/install` directory on the CD.

The syntax to run backup_oid.sh is:

```
backup_oid.sh -connect net_service_name -pass password_for_DB_account_'ods'
```

The backup_oid.sh script does the following:

- Exports Oracle Internet Directory schema. As it does this, it generates `.dmp` files—for example, `attr_store.dmp`—in `$`*`ORACLE_HOME/`*`ldap/load` directory
- Backs up the Oracle Internet Directory subtree by using the ldifwrite utility. As it does this, it generates the file `OID_userdata.ldif` in `$`*`ORACLE_HOME/`*`ldap/load`. The subtree under `cn=OracleSchemaVersion` (if it exists) is also backed up as `orcl_schemaver.ldif` in the `$`*`ORACLE_HOME/`*`ldap/load` directory.

If you plan to install Oracle Internet Directory release 3.0.1 in the same *ORACLE_HOME*, then save these generated files in some other location.

Task 2: Perform a Fresh Installation of Oracle Internet Directory Release 3.0.1

See Also: Installation documentation for your operating system

Task 3: Restore the User-Defined Schema and Data from the Previous Version of Oracle Internet Directory

To do this:

1. Make sure that the directory server is not running.

2. Copy the following files to `$ORACLE_HOME/1dap/load`:
   - Backed up Oracle Internet Directory schema dump files—that is, files with the extension `.dmp`
   - The file `OID_userdata.ldif`
3. Run the script restore_oid.sh located in `$ORACLE_HOME/ldap/install`.

   The syntax for restore_oid.sh is:

   ```
   restore_oid.sh -connect net_service_name -pass password_for_DB_account_‘ods'
   ```

The restore_oid.sh script does the following:

- Imports the Oracle Internet Directory schema from the dump files
- Inserts the schema differences between the previous release and release 3.0.1
- Bulkloads the data from the LDIF file with the `-restore` option

Task 4: Start Oracle Internet Directory Processes

Start OID Monitor and the directory server.

See Also:

- "Starting the OID Monitor" on page 4-2
- "Starting an Oracle Directory Server Instance" on page 4-4

F

Migrating Data from Other LDAP-Compliant Directories

This appendix tells how to migrate data from LDAP Version 3-compatible directories into Oracle Internet Directory.

This appendix contains these topics:

- About the Data Migration Process
- Migrating Data

About the Data Migration Process

You can import data from a third-party LDAP-compliant directory into Oracle Internet Directory by saving the data in an LDIF file. LDIF is the IETF-sanctioned ASCII interchange format for representing LDAP-compliant directory data as a file. All LDAP-compliant directories should be able to export their contents into one or more LDIF files representing the DIT at the time of export.

Be aware that certain proprietary attributes or metadata may be included in a given product's LDIF output. You must remove this extraneous data from the LDIF file before you import the file into Oracle Internet Directory. In such cases, you need to perform some additional steps before importing the LDIF files into Oracle Internet Directory. The next section explains these steps.

> **See Also:** The LDIF technical specification available for download at: `http://www.ietf.org/rfc/rfc2849.txt`

Migrating Data

This section contains these topics:

- Task 1: Export Data from the Non-Oracle Internet Directory Server into LDIF File Format
- Task 2: Analyze the LDIF User Data for Any Required Schema Additions Referenced in the LDIF Data
- Task 3: Extend the Schema in Oracle Internet Directory
- Task 4: Remove Any Proprietary Directory Data from the LDIF File
- Task 5: Remove Operational Attributes from the LDIF File
- Task 6: Remove Incompatible userPassword Attribute Values from the LDIF File
- Task 7: Run the bulkload.sh -check Mode and Determine Any Remaining Schema Violations or Duplication Errors

Task 1: Export Data from the Non-Oracle Internet Directory Server into LDIF File Format

See the vendor-supplied documentation for instructions. If flags or options exist for exporting data from the foreign directory, be sure to select the method that:

- Produces LDIF output with the least amount of proprietary information included
- Provides maximum conformance to the IETF Request for Comments 2849 mentioned in About the Data Migration Process on page F-2

Task 2: Analyze the LDIF User Data for Any Required Schema Additions Referenced in the LDIF Data

Any attributes not found in the Oracle Internet Directory base schema require extension of the Oracle Internet Directory base schema prior to the importation of the LDIF file. Some directories may support the use of configuration files for defining extensions to their base schema (Oracle Internet Directory does not). If you have a configuration file you can use it as a guideline for extending the base schema in Oracle Internet Directory in "Task 3: Extend the Schema in Oracle Internet Directory".

Task 3: Extend the Schema in Oracle Internet Directory

See Chapter 7, "Managing the Directory Schema" for tips on how to extend the directory schema in Oracle Internet Directory. You can do this by using either Oracle Directory Manager or command-line tools.

Task 4: Remove Any Proprietary Directory Data from the LDIF File

Certain elements of the LDAP v3 standard have not yet been formalized, such as **ACI** attributes. As a result, various directory vendors implement ACI policy objects in ways that do not translate well across vendor installations.

After the basic entry data has been imported from the cleaned up LDIF file to Oracle Internet Directory, you must explicitly reapply security policies in the Oracle Internet Directory environment. You can do this by using either Oracle Directory Manager, or command-line tools and LDIF files containing the desired **ACP** information.

There may be other proprietary metadata unrelated to access control. You should remove this as well. Understanding the various IETF RFCs can help you determine which directory metadata is proprietary to a given vendor and which complies with the LDAP standards, and is thus portable by way of an LDIF file.

Task 5: Remove Operational Attributes from the LDIF File

Four of the standard LDAP v3 operational attributes, namely, `creatorsName`, `createTimestamp`, `modifiersName`, and `modifyTimestamp` are automatically generated by Oracle Internet Directory whenever entries are created or imported. It is not possible to instantiate these values from existing directory data, for example by using LDIF file importation. Therefore you should remove these attributes from the file before attempting to import.

Task 6: Remove Incompatible userPassword Attribute Values from the LDIF File

Oracle Internet Directory release 3.0.1 supports the following `userPassword` attribute hash algorithms:

- No encryption
- **MD4**
- **MD5**
- **SHA**
- **UNIX Crypt**

The `userPassword` attribute hash values used by some vendor products are not compatible with Oracle Internet Directory. As a result, you must remove all lines corresponding to the `userPassword` attribute and value from the LDIF data file unless they are represented in plain text or contain no value. After importation of the LDIF data, you must re-enter manually or upload hashed `userPassword` information separately into the directory.

Task 7: Run the bulkload.sh -check Mode and Determine Any Remaining Schema Violations or Duplication Errors

Before generating and loading an LDIF file, always perform a check on it by using the bulkload utility check mode. The bulkload output reports any inconsistencies in the data.

> **See Also:** "bulkload Syntax" on page A-28 for instructions on how to use the bulkload check mode

G

Troubleshooting

This appendix explains typical problems that you could encounter while running or installing Oracle Internet Directory. It contains these topics:

- Installation Errors
- Administration Error Messages and Causes

Installation Errors

During installation and configuration of the Oracle9*i* database server, you must select the character set UTF-8. If you select any other character set, the directory server will not function properly.

Administration Error Messages and Causes

This section contains a list of all the Oracle directory server error messages that you can encounter. Each message is followed by its most probable causes.

This section contains these topics:

- Oracle Database Server Error Due to Schema Modifications
- Standard Error Messages Returned from Oracle Directory Server
- Additional Error Messages

Oracle Database Server Error Due to Schema Modifications

ORA-1562

Cause: If you attempt to add more schema components than can fit in the rollback segment space, you will encounter this error and the modifications will not commit. To solve this, increase the size of the rollback segments in the database server.

Standard Error Messages Returned from Oracle Directory Server

The following are standard error messages. Oracle Internet Directory also returns other messages listed and described in "Additional Error Messages" on page G-6.

00—LDAP_SUCCESS

Cause: The operation was successful.

01—LDAP_OPERATIONS_ERROR

Cause: General errors encountered by the server when processing the request.

02—LDAP_PROTOCOL_ERROR

Cause: The client request did not meet the LDAP protocol requirements, such as format or syntax. This can occur in the following situations:

- Server encounters a decoding error while parsing the incoming request
- The request is an add or modify request that specifies the addition of an attribute type to an entry but no values specified

- Error reading SSL credentials
- An unknown type of modify operation is specified (other than LDAP_MOD_ADD, LDAP_MOD_DELETE, and LDAP_MOD_REPLACE)
- Unknown search scope

03—LDAP_TIMELIMIT_EXCEEDED

Cause: Search took longer than the time limit specified. If you have not specified a time limit for the search, Oracle Internet Directory uses a default time limit of one hour.

04—LDAP_SIZELIMIT_EXCEEDED

Cause: More entries match the search query than the size limit specified. If you have not specified a size limit for the search, Oracle Internet Directory uses a default size limit.

05—LDAP_COMPARE_FALSE

Cause: Presented value is not the same as the one in the entry.

06—LDAP_COMPARE_TRUE

Cause: Presented value is same as the one in the entry.

07—LDAP_STRONG_AUTH_NOT_SUPPORTED

Cause: Bind method is not supported by the server.

08—LDAP_STRONG_AUTH_REQUIRED

Cause: Strong authentication is required. Oracle Internet Directory does not return this message at the present time.

09—LDAP_PARTIAL_RESULTS

Cause: Server returned a referral.

10—LDAP_REFERRAL

Cause: Server returned a referral.

11—LDAP_ADMINLIMIT_EXCEEDED

Cause: Oracle Internet Directory does not return this message at the present time.

12—LDAP_UNAVAILABLE_CRITICALEXTENSION

Cause: Specified request is not supported

16—LDAP_NO_SUCH_ATTRIBUTE

Cause: Attribute does not exist in the entry specified in the request.

17—LDAP_UNDEFINED_TYPE

Cause: Specified attribute type is undefined in the schema.

18—LDAP_INAPPROPRIATE_MATCHING

Cause: Specified matching rule is inappropriate for the attribute type. Oracle Internet Directory does not return this message at the present time.

19—LDAP_CONSTRAINT_VIOLATION

Cause: The value in the request violated certain constraints.

20—LDAP_TYPE_OR_VALUE_EXISTS

Cause: Duplicate values specified for the attribute.

21—LDAP_INVALID_SYNTAX

Cause: Specified *attribute* syntax is invalid. In a search, the *filter* syntax is invalid.

32—LDAP_NO_SUCH_OBJECT

Cause: The base specified for the operation does not exist.

33—LDAP_ALIAS_PROBLEM

Cause: Oracle Internet Directory does not return this message at the present time.

34—LDAP_INVALID_DN_SYNTAX

Cause: Error in the DN syntax.

35—LDAP_IS_LEAF

Cause: The entry is a leaf (terminal entry). Oracle Internet Directory does not return this message at the present time.

36—LDAP_ALIAS_DEREF_PROBLEM

Cause: Oracle Internet Directory does not return this message at the present time.

48—LDAP_INAPPROPRIATE_AUTH

Cause: Oracle Internet Directory does not return this message at the present time.

49—LDAP_INVALID_CREDENTIALS

Cause: Bind failed because the credentials are not correct.

50—LDAP_INSUFFICIENT_ACCESS

Cause: The client does not have access to perform this operation.

51—LDAP_BUSY

Cause: Server cannot accept any more client connections. Oracle Internet Directory does not return this message at the present time.

52—LDAP_UNAVAILABLE

Cause: Cannot contact the server at all. Oracle Internet Directory does not return this message at the present time.

53—LDAP_UNWILLING_TO_PERFORM

Cause: General error, or server is in read-only mode.

54—LDAP_LOOP_DETECT

Cause: Oracle Internet Directory does not return this message at the present time.

64—LDAP_NAMING_VIOLATION

Cause: Oracle Internet Directory does not return this message at the present time.

65—LDAP_OBJECT_CLASS_VIOLATION

Cause: A change to the entry violates the objectclass definition.

66— LDAP_NOT_ALLOWED_ON_NONLEAF

Cause: The entry to be deleted has children.

67—LDAP_NOT_ALLOWED_ON_RDN

Cause: Cannot perform the operation on RDN attributes—for example, you cannot delete the RDN attribute of the entry.

68—LDAP_ALREADY_EXISTS

Cause: Duplicate ADD condition.

69—LDAP_NO_OBJECT_CLASS_MODS

Cause: Oracle Internet Directory does not return this message at the present time.

70—LDAP_RESULTS_TOO_LARGE

Cause: Oracle Internet Directory does not return this message at the present time.

80—LDAP_OTHER

Cause: Oracle Internet Directory does not return this message at the present time.

81—LDAP_SERVER_DOWN

Cause: Can’t contact LDAP server. This message is returned from the SDK.

82—LDAP_LOCAL_ERROR

Cause: The client encountered an internal error. This message is returned from the client SDK.

83—LDAP_ENCODING_ERROR

Cause: The client encountered an error in encoding the request. This message is returned from the SDK.

84—LDAP_DECODING_ERROR

Cause: The client encountered an error in decoding the request. This message is returned from the SDK.

85—LDAP_TIMEOUT

Cause: Client encountered the time-out specified for the operation. This message is returned from the SDK.

86—LDAP_AUTH_UNKNOWN

Cause: Authentication method is unknown to the client SDK.

87—LDAP_FILTER_ERROR

Cause: Bad search filter

88—LDAP_USER_CANCELLED

Cause: User cancelled operation

89—LDAP_PARAM_ERROR

Cause: Bad parameter to an LDAP routine

90—LDAP_NO_MEMORY

Cause: Out of memory

Additional Error Messages

These messages do not display error codes.

The Oracle Internet Directory application replaces the *parameter* tag seen in some of the messages below with the appropriate run-time value.

%s attribute not found.

Cause: The particular attribute type is not defined in the schema.

<parameter> not found for attribute <parameter>.

Cause: Value not found in the attribute. (ldapmodify)

Admin domain does not contain schema information for objectclass <parameter>.

Cause: The object class specified in the request is not present in the schema.

Attempted to add a Class with oid <parameter> taken by other class.

Cause: Duplicate object identifier specified. (schema modification)

Attribute <parameter> already in use.

Cause: Duplicate attribute name. (schema modification)

Attribute <parameter> has syntax error.

Cause: Syntax error in the attribute name definition. (schema modification)

Attribute <parameter> is not supported in the schema.

Cause: Attribute not defined. (all operations)

Attribute <parameter> is single valued.

Cause: Attribute is single-valued. (ldapadd & ldapmodify)

Attribute <parameter> not present in the entry.

Cause: This attribute does not exist in the entry. (ldapmodify)

Bad attribute definition.

Cause: Syntax error in attribute definition. (schema modification)

Currently Not Supported

Cause: The version of LDAP request is not supported by this server.

Entry to be deleted not found.

Cause: DN specified in the delete operation not found.

Entry to be modified not found

Cause: The entry specified in the request is not found.

Error encountered while adding <parameter> to the entry

Cause: Returned when modify add operation is invoked. A possible cause is that the system resource is unavailable.

Error encountered while encrypting an attribute value.

Cause: Error in encrypting user password. (all operations)

Error in DN Normalization.

Cause: DN specified is invalid. Syntax error encountered in parsing the DN. (all operations)

Error in hashing <parameter> attribute.

Cause: Error in creating hash entry for the attribute. (schema modification)

Error in hashing <parameter> objectclass.

Cause: Error in creating hash entry for the objectclass. (schema modification)

Error in Schema hash creation.

Cause: Error while creating hash table for schema. (schema modification)

Error replacing <parameter>.

Cause: Error in replacing this attribute. (ldapmodify)

Error while normalizing value for attribute <parameter>.

Cause: Error in normalizing value for the attribute. (all operations)

Failed to find <parameter> in mandatory or optional attribute list.

Cause: Attribute specified does not exist in either the mandatory or optional attribute list as required by the object class(es).

Function Not Implemented

Cause: The feature / request is currently not supported.

INVALID ACI is <parameter>

Cause: The particular ACI you specified in a request is invalid.

Mandatory attribute <parameter> is not defined in Admin Domain <parameter>.

Cause: MUST refers to attribute not defined. (schema modification)

Mandatory Attribute missing.

Cause: The mandatory attribute for the particular entry is missing, as required by the particular object class.

Matching rule, <parameter>, not defined.

Cause: Matching rule not defined in the server. (schema modification)

MaxConn Reached

Cause: The maximum number of concurrent connections to the LDAP server has been reached.

Modifying the Naming attribute for the entry without modifying the DN.

Cause: Cannot modify the naming attributes using ldap_modify. A naming attribute, such as *cn* is an element in the DN.

New Parent not found.

Cause: New parent specified in modifydn operation does not exist.(ldapmodifydn)

Object already exists.

Cause: Duplicate entry. (ldapadd and ldapmodifydn)

Object ID <parameter> already in use.

Cause: Duplicate object identifier specified. (schema modification)

Objectclass <parameter> already in use. m

Cause: Duplicate Objectclass name. (schema modification)

Objectclass attribute missing.

Cause: The objectclass attribute is missing for this particular entry.

OID <parameter> has syntax error.

Cause: syntax error in the object identifier definition. (schema modification)

One of the attributes in the entry has duplicate value

Cause: You entered two values for the same attribute in the entry you are creating.

Operation not allowed on the <parameter>.

Cause: Operation not allowed on this entry. (modify, add, and delete)

Operation not allowed on the DSE Entry.

Cause: Can't do this operation on DSE entry. (delete)

Optional attribute <parameter> is not defined in Admin Domain <parameter>.

Cause: MAY refers to attribute not defined. (schema modification)

Parent entry not found in the directory.

Cause: Parent entry does not exist. (ldapadd and perhaps ldapmodifydn)

Super object <parameter> is not defined in Admin Domain <parameter>.

Cause: SUP types refer to non-existing class. (schema modification)

Super type undefined.

Cause: SUP type does not exist. (schema modification)

Super user addition not permitted.

Cause: Cannot create super user entry. (ldapadd)

Syntax, <parameter>, not defined.

Cause: Syntax not defined in the server. (schema modification)

The attribute or the value specified in the RDN does not exist in the entry.

Cause: AVA specified as the RDN does not exist in the entry. (ldapadd)

Unknown search scope

Cause: The search scope specified in the LDAP request is not recognized.

Version Not Supported

Cause: The version of the LDAP request is not supported by this server.

Glossary

access control item (ACI)

An attribute that determines who has what type of access to what directory data. It contains a set of rules for structural access items, which pertain to entries, and content access items, which pertain to attributes. Access to both structural and content access items may be granted to one or more users or groups.

access control list (ACL)

The group of access directives that you define. The directives grant levels of access to specific data for specific clients, or groups of clients, or both.

access control policy point

An entry that contains security directives that apply downward to all entries at lower positions in the **directory information tree (DIT)**.

ACI

See **access control item (ACI)**.

ACL

See **access control list (ACL)**.

ACP

See **access control policy point**.

administrative area

A subtree on a directory server whose entries are under the control (schema, ACL, and collective attributes) of a single administrative authority.

advanced symmetric replication (ASR)

See **Oracle9i Replication**

agent

See **directory integration agent**

agent profile

In an Oracle Directory Integration platform environment, an entry in Oracle Internet Directory that specifies:

- Configuration parameters for integration agents
- Mapping rules for synchronizing between a connected directory and Oracle Internet Directory

anonymous authentication

The process by which the directory authenticates a user without requiring a user name and password combination. Each anonymous user then exercises the privileges specified for anonymous users.

API

See **application program interface**.

application program interface

Programs to access the services of a specified application. For example, LDAP-enabled clients access directory information through programmatic calls available in the LDAP API.

ASR

See **Oracle9i Replication**

attribute

An item of information that describes some aspect of an entry. An entry comprises a set of attributes, each of which belongs to an **object class**. Moreover, each attribute has both a *type*, which describes the kind of information in the attribute, and a *value*, which contains the actual data.

attribute configuration file

In an Oracle Directory Integration platform environment, a file that specifies attributes of interest in a connected directory.

attribute type

The kind of information an attribute contains, for example, `jobTitle`.

attribute value

The particular occurrence of information appearing in that entry. For example, the value for the `jobTitle` attribute could be `manager`.

authentication

The process of verifying the identity of a user, device, or other entity in a computer system, often as a prerequisite to allowing access to resources in a system.

authorization

Permission given to a user, program, or process to access an object or set of objects.

binding

The process of authenticating to a directory.

central directory

In an Oracle Directory Integration platform environment, the directory that acts as the central repository. In an Oracle Directory Integration platform environment, Oracle Internet Directory is the central directory.

certificate

An ITU x.509 v3 standard data structure that securely binds an identity to a public key. A certificate is created when an entity's public key is signed by a trusted identity: a **certificate authority (CA)**. This certificate ensures that the entity's information is correct and that the public key actually belongs to that entity.

certificate authority (CA)

A trusted third party that certifies that other entities—users, databases, administrators, clients, servers—are who they say they are. The certificate authority verifies the user's identity and grants a certificate, signing it with the certificate authority's private key.

certificate chain

An ordered list of certificates containing an end-user or subscriber certificate and its certificate authority certificates.

change logs

A database that records changes made to a directory server.

cipher suite

In SSL, a set of authentication, encryption, and data integrity algorithms used for exchanging messages between network nodes. During an SSL handshake, the two nodes negotiate to see which cipher suite they will use when transmitting messages back and forth.

cold backup

The procedure to add a new node to an existing replicating system by using the database copy procedure.

concurrency

The ability to handle multiple requests simultaneously. Threads and processes are examples of concurrency mechanisms.

concurrent clients

The total number of clients that have established a session with Oracle Internet Directory.

concurrent operations

The number of operations that are being executed on the directory from all of the concurrent clients. Note that this is not necessarily the same as the concurrent clients, because some of the clients may be keeping their sessions idle.

configset

See **configuration set entry**.

configuration set entry

A directory entry holding the configuration parameters for a specific instance of the directory server. Multiple configuration set entries can be stored and referenced at run-time. The configuration set entries are maintained in the subtree specified by the subConfigsubEntry attribute of the DSE, which itself resides in the associated **directory information base (DIB)** against which the servers are started.

connect descriptor

A specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information.

The destination service is indicated by using its service name for Oracle9*i* release 9.0.1 database or its Oracle System Identifier (SID) for Oracle release 8.0 or version 7 databases. The network route provides, at a minimum, the location of the listener through use of a network address.

connected directory

In an Oracle Directory Integration platform environment, any directory or repository other than the **central directory**. In such an environment, Oracle Internet Directory serves as the central directory, and all other directories are connected directories. Synchronization always happens between Oracle Internet Directory and a connected directory.

consumer

A directory server that is the destination of replication updates. Sometimes called a slave.

contention

Competition for resources.

context prefix

The **DN** of the root of a **naming context**.

cryptography

The practice of encoding and decoding data, resulting in secure messages.

data integrity

The guarantee that the contents of the message received were not altered from the contents of the original message sent.

decryption

The process of converting the contents of an encrypted message (ciphertext) back into its original readable format (plaintext).

default knowledge reference

A **knowledge reference** that is returned when the base object is not in the directory, and the operation is performed in a naming context not held locally by the server. A default knowledge reference typically sends the user to a server that has more knowledge about the directory partitioning arrangement.

DES

Data Encryption Standard, a block cipher developed by IBM and the U.S. government in the 1970's as an official standard.

DIB

See **directory information base (DIB)**.

directory information base (DIB)

The complete set of all information held in the directory. The DIB consists of entries that are related to each other hierarchically in a **directory information tree (DIT).**

directory information tree (DIT)

A hierarchical tree-like structure consisting of the DNs of the entries.

directory integration agent

In an Oracle Directory Integration platform environment, a program that interacts with a connected directory to synchronize changes between the connected directory and Oracle Internet Directory.

directory integration profile

In an Oracle Directory Integration platform environment, an entry in Oracle Internet Directory that contains configuration information required for synchronization.

directory integration server

In an Oracle Directory Integration platform environment, the server that drives the synchronization of data between Oracle Internet Directory and a **connected directory**.

directory naming context

See **naming context**.

directory replication group (DRG)

The directory servers participating in a replication agreement.

directory server instance

A discrete invocation of a directory server. Different invocations of a directory server, each started with the same or different configuration set entries and startup flags, are said to be different directory server instances.

directory-specific entry (DSE)

An entry specific to a **directory system agent (DSA)**. Different DSAs may hold the same DIT name, but have different contents—that is, the contents can be specific to the DSA holding it. A DSE is an entry with contents specific to the DSA holding it.

directory system agent (DSA)

The X.500 term for a directory server.

distinguished name (DN)

The unique name of a directory entry. It comprises all of the individual names of the parent entries back to the root.

DIS

See **directory integration server**

DIT

See **directory information tree (DIT)**

DN

See **distinguished name (DN)**

DRG

See **directory replication group (DRG)**

DSA

See **directory system agent (DSA)**

DSE

See **directory-specific entry (DSE)**

encryption

The process of disguising the contents of a message and rendering it unreadable (ciphertext) to anyone but the intended recipient.

entry

The building block of a directory, it contains information about an object of interest to directory users.

export agent

In an Oracle Directory Integration platform environment, an agent that exports data out of Oracle Internet Directory.

export data file

In an Oracle Directory Integration platform environment, the file that contains data exported by an **export agent**.

export file

See **export data file**.

external agent

A directory integration agent that is independent of the Oracle Directory Integration server. The Oracle directory integration server does not provide scheduling, mapping, or error handling services for it. An external agent is typically used when a third party metadirectory solution is integrated with the Oracle Directory Integration platform.

failover

The process of failure recognition and recovery.

filter

A method of qualifying data, usually data that you are seeking. Filters are always expressed as DNs, for example: `cn=susie smith, o=acme, c=us`.

global unique identifier (GUID)

In a multi-master replication environment, an entry replicated on multiple nodes has the same DN on each node. However, even though it has the same DN, it is assigned a different GUID on each node. For example, the same DN can be replicated on both node1 and node2, but the GUID for that DN as it resides on node1 would be different from the GUID for that DN on node2.

grace login

A login occurring within the specified period before password expiration.

guest user

One who is not an anonymous user, and, at the same time, does not have a specific user entry.

GUID

See **global unique identifier (GUID)**.

handshake

A protocol two computers use to initiate a communication session.

hash

A number generated from a string of text with an algorithm. The hash value is substantially smaller than the text itself. Hash numbers are used for security and for faster access to data.

import agent

In an Oracle Directory Integration platform environment, an agent that imports data into Oracle Internet Directory.

import file

In an Oracle Directory Integration platform environment, the file containing the data imported by an **import agent**.

inherit

When an object class has been derived from another class, it also derives, or inherits, many of the characteristics of that other class. Similarly, an attribute subtype inherits the characteristics of its supertype.

instance

See **directory server instance**.

integration agent

See **agent**.

integrity

The guarantee that the contents of the message received were not altered from the contents of the original message sent.

Internet Engineering Task Force (IETF)

The principal body engaged in the development of new Internet standard specifications. It is an international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet.

Internet Message Access Protocol (IMAP)

A protocol allowing a client to access and manipulate electronic mail messages on a server. It permits manipulation of remote message folders, also called mailboxes, in a way that is functionally equivalent to local mailboxes.

key

A string of bits used widely in cryptography, allowing people to encrypt and decrypt data; a key can be used to perform other mathematical operations as well. Given a cipher, a key determines the mapping of the plaintext to the ciphertext.

key pair

A **public key** and its associated **private key**.

See **public/private key pair**.

knowledge reference

The access information (name and address) for a remote **DSA** and the name of the **DIT** subtree that the remote DSA holds. Knowledge references are also called referrals.

latency

The time a client has to wait for a given directory operation to complete. Latency can be defined as wasted time. In networking discussions, latency is defined as the travel time of a packet from source to destination.

LDAP

See **Lightweight Directory Access Protocol (LDAP)**.

LDIF

See **LDAP Data Interchange Format (LDIF)**.

Lightweight Directory Access Protocol (LDAP)

A standard, extensible directory access protocol. It is a common language that LDAP clients and servers use to communicate. The framework of design conventions supporting industry-standard directory products, such as the Oracle Internet Directory.

LDAP Data Interchange Format (LDIF)

The set of standards for formatting an input file for any of the LDAP command-line utilities.

man-in-the-middle

A security attack characterized by the third-party, surreptitious interception of a message. The third-party, the *man-in-the-middle*, decrypts the message, re-encrypts it (with or without alteration of the original message), and retransmits it to the originally-intended recipient—all without the knowledge of the legitimate sender and receiver. This type of security attack works only in the absence of **authentication**.

mapping rules file

In an Oracle Directory Integration platform environment, the file that specifies mappings between Oracle Internet Directory attributes and those in a **connected directory**.

master definition site (MDS)

In replication, a master definition site is the Oracle Internet Directory database from which the administrator runs the configuration scripts.

master site

In replication, a master site is any site other than the master definition site that participates in LDAP replication.

matching rule

In a search or compare operation, determines equality between the attribute value sought and the attribute value stored. For example, matching rules associated with the `telephoneNumber` attribute could cause "(650) 123-4567" to be matched with either "(650) 123-4567" or "6501234567" or both. When you create an attribute, you associate a matching rule with it.

MD4

A one-way hash function that produces a 128-bit hash, or message digest. If as little as a single bit value in the file is modified, the MD4 checksum for the file will change. Forgery of a file in a way that will cause MD4 to generate the same result as that for the original file is considered extremely difficult.

MD5

An improved version of MD4.

MDS

See **master definition site (MDS)**.

metadirectory

A directory solution that shares information between all enterprise directories, integrating them into one virtual directory. It centralizes administration, thereby reducing administrative costs. It synchronizes data between directories, thereby ensuring that it is consistent and up-to-date across the enterprise.

native agent

In an Oracle Directory Integration platform environment, an **agent** that runs under the control of the **directory integration server**.

naming attribute

A specialized attribute that holds values for different types of **RDN**. A naming attribute is identifiable by its mnemonic label, usually `cn`, `sn`, `ou`, `o`, `c`, and so on. For example, the naming attribute `c` is the mnemonic for the naming attribute `country`, and it holds the RDN for specific country values.

naming context

A subtree that resides entirely on one server. It must be contiguous, that is, it must begin at an entry that serves as the top of the subtree, and extend downward to either leaf entries or **knowledge references** (also called referrals) to subordinate naming contexts. It can range in size from a single entry to the entire DIT.

Oracle Net Services

The foundation of the Oracle family of networking products, allowing services and their client applications to reside on different computers and communicate. The main function of Oracle Net Services is to establish network sessions and transfer data between a client application and a server. Oracle Net Services is located on each computer in the network. Once a network session is established, Oracle Net Services acts as a data courier for the client and the server.

net service name

A simple name for a service that resolves to a connect descriptor. Users initiate a connect request by passing a user name and password along with a net service name in a connect string for the service to which they wish to connect:

```
CONNECT username/password@net_service_name
```

Depending on your needs, net service names can be stored in a variety of places, including:

- Local configuration file, tnsnames.ora, on each client

- Directory server
- Oracle Names server
- External naming service, such as NDS, NIS or CDS

object class

A named group of attributes. When you want to assign attributes to an entry, you do so by assigning to that entry the object classes that hold those attributes.

All objects associated with the same object class share the same attributes.

OEM

See **Oracle Enterprise Manager**.

OID Control Utility

A command-line tool for issuing run-server and stop-server commands. The commands are interpreted and executed by the **OID Monitor** process.

OID Database Password Utility

The utility used to change the password with which Oracle Internet Directory connects to an Oracle database.

OID Monitor

The Oracle Internet Directory component that initiates, monitors, and terminates the Oracle directory server processes. It also controls the replication server if one is installed, and the Oracle directory integration server.

one-way function

A function that is easy to compute in one direction but quite difficult to reverse compute, that is, to compute in the opposite direction.

one-way hash function

A **one-way function** that takes a variable sized input and creates a fixed size output.

Oracle Call Interface (OCI)

An application programming interface (API) that enables you to create applications that use the native procedures or function calls of a third-generation language to access an Oracle database server and control all phases of SQL statement execution.

Oracle Directory Integration platform

A component of **Oracle Internet Directory**. It allows various information repositories to synchronize with Oracle Internet Directory and to form a single virtual directory.

Oracle directory integration server (DIS)

In an Oracle Directory Integration platform environment, the server that drives the synchronization of data between Oracle Internet Directory and a **connected directory**.

Oracle Directory Manager

A Java-based tool with a graphical user interface for administering Oracle Internet Directory.

Oracle Enterprise Manager

A separate Oracle product that combines a graphical console, agents, common services, and tools to provide an integrated and comprehensive systems management platform for managing Oracle products.

Oracle Internet Directory

A general purpose directory service that enables retrieval of information about dispersed users and network resources. It combines Lightweight Directory Access Protocol (LDAP) Version 3 with the high performance, scalability, robustness, and availability of Oracle9*i*.

Oracle PKI certificate usages

Defines Oracle application types that a **certificate** supports.

Oracle Wallet Manager

A Java-based application that security administrators use to manage public-key security credentials on clients and servers.

Oracle9*i* Replication

A feature in Oracle9*i* that allows database tables to be kept synchronized across two Oracle databases.

other information repository

In an Oracle Directory Integration platform environment, in which Oracle Internet Directory serves as the **central directory**, any information repository except Oracle Internet Directory.

partition

A unique, non-overlapping directory naming context that is stored on one directory server.

partner agent

A directory integration agent for which the Oracle Directory Integration server performs mapping, scheduling, and error handling.

PKCS #12

A **public-key encryption** standard (PKCS). RSA Data Security, Inc. PKCS #12 is an industry standard for storing and transferring personal authentication credentials—typically in a format called a **wallet**.

plaintext

Message text that has not been encrypted.

private key

In public-key cryptography, this key is the secret key. It is primarily used for decryption but is also used for encryption with digital signatures.

profile

See **directory integration profile**

proxy user

A kind of user typically employed in an environment with a middle tier such as a firewall. In such an environment, the end user authenticates to the middle tier. The middle tier then logs into the directory on the end user's behalf, but does so as a proxy user. A proxy user has the privilege to switch identities and, once it has logged into the directory, switches to the end user's identity. It then performs operations on the end user's behalf, using the authorization appropriate to that particular end user.

public key

In public-key cryptography this key is made public to all, it is primarily used for encryption but can be used for verifying signatures.

public-key cryptography

Cryptography based on methods involving a public key and a private key.

public-key encryption

The process in which the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the message is decrypted by the recipient using the recipient's private key.

public/private key pair

A mathematically related set of two numbers where one is called the private key and the other is called the public key. Public keys are typically made widely available, while private keys are available only to their owners. Data encrypted with a public key can only be decrypted with its associated private key and vice versa. Data encrypted with a public key cannot be decrypted with the same public key.

referral

See **knowledge reference**.

relational database

A structured collection of data that stores data in tables consisting of one or more rows, each containing the same set of columns. Oracle makes it very easy to link the data in multiple tables. This is what makes Oracle a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the tables. The link is based on one or more fields common to both tables.

replica

Each copy of a naming context that is contained within a single server.

RDN

See **relative distinguished name (RDN).**

registry entries

Entries containing run-time information associated with invocations of Oracle Internet Directory servers, called **directory server instances**. Registry entries are stored in the directory itself, and remain there until the corresponding directory server instance stops.

relative distinguished name (RDN)

The local, most granular level entry name. It has no other qualifying entry names that would serve to uniquely address the entry. In the example, `cn=Smith,o=acme,c=US`, the RDN is `cn=Smith`.

replication agreement

A special directory entry that represents the replication relationship among the directory servers in a **directory replication group (DRG)**.

response time

The time between the submission of a request and the completion of the response.

root DSE

See **root directory specific entry**.

root directory specific entry

An entry storing operational information about the directory. The information is stored in a number of attributes.

SASL

See **Simple Authentication and Security Layer (SASL)**

scalability

The ability of a system to provide throughput in proportion to, and limited only by, available hardware resources.

schema

The collection of **attributes**, **object classes**, and their corresponding matching rules.

Secure Hash Algorithm (SHA)

An algorithm that takes a message of less than 264 bits in length and produces a 160-bit message digest. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.

Secure Socket Layer (SSL)

An industry standard protocol designed by Netscape Communications Corporation for securing network connections. SSL provides authentication, encryption, and data integrity using public key infrastructure (PKI).

service time

The time between the initiation of a request and the completion of the response to the request.

session key

A key for symmetric-key cryptosystems that is used for the duration of one message or communication session

SGA

See **System Global Area (SGA)**.

SHA

See **Secure Hash Algorithm (SHA)**.

shared server

A server that is configured to allow many user processes to share very few server processes, so the number of users that can be supported is increased. With shared server configuration, many user processes connect to a dispatcher. The dispatcher directs multiple incoming network session requests to a common queue. An idle shared server process from a shared pool of server processes picks up a request from the queue. This means a small pool of server processes can server a large amount of clients. Contrast with dedicated server.

sibling

An entry that has the same parent as one or more other entries.

simple authentication

The process by which the client identifies itself to the server by means of a DN and a password which are not encrypted when sent over the network. In the simple authentication option, the server verifies that the DN and password sent by the client match the DN and password stored in the directory.

Simple Authentication and Security Layer (SASL)

A method for adding authentication support to connection-based protocols. To use this specification, a protocol includes a command for identifying and authenticating a user to a server and for optionally negotiating a security layer for subsequent protocol interactions. The command has a required argument identifying a SASL mechanism.

single key-pair wallet

A **PKCS #12**-format **wallet** that contains a single user **certificate** and its associated **private key**. The **public key** is imbedded in the certificate.

slave

See **consumer**.

SLAPD

Standalone LDAP daemon.

smart knowledge reference

A **knowledge reference** that is returned when the knowledge reference entry is in the scope of the search. It points the user to the server that stores the requested information.

specific administrative area

Administrative areas control:

- Subschema administration
- Access control administration
- Collective attribute administration

A *specific* administrative area controls one of the above aspects of administration. A specific administrative area is part of an autonomous administrative area.

sponsor node

In replication, the node that is used to provide initial data to a new node.

SSL

See **Secure Socket Layer (SSL)**.

subclass

An object class derived from another object class. The object class from which it is derived is called its **superclass**.

subschema DN

The list of DIT areas having independent schema definitions.

subentry

A type of entry containing information applicable to a group of entries in a subtree. The information can be of these types:

- **access control policy points**
- Schema rules

- Collective attributes

Subentries are located immediately below the root of an administrative area.

subordinate reference

A knowledge reference pointing downward in the DIT to a naming context that starts immediately below an entry.

subtype

An attribute with one or more options, in contrast to that same attribute without the options. For example, a `commonName` (`cn`) attribute with American English as an option is a subtype of the `commonName` (`cn`) attribute without that option. Conversely, the `commonName` (`cn`) attribute without an option is the **supertype** of the same attribute with an option.

subACLSubentry

A specific type of subentry that contains ACL information.

subSchemaSubentry

A specific type of **subentry** containing schema information.

super user

A special directory administrator who typically has full access to directory information.

superclass

The object class from which another object class is derived. For example, the object class `person` is the superclass of the object class `organizationalPerson`. The latter, namely, `organizationalPerson`, is a **subclass** of `person` and **inherits** the attributes contained in `person`.

superior reference

A knowledge reference pointing upward to a DSA that holds a naming context higher in the DIT than all the naming contexts held by the referencing DSA.

supertype

An attribute without options, in contrast to the same attribute with one or more options. For example, the `commonName` (`cn`) attribute without an option is the supertype of the same attribute with an option. Conversely, a `commonName` (`cn`) attribute with American English as an option is a **subtype** of the commonName (cn) attribute without that option.

supplier

In replication, the server that holds the master copy of the naming context. It supplies updates from the master copy to the **consumer** server.

System Global Area (SGA)

A group of shared memory structures that contain data and control information for one Oracle database instance. If multiple users are concurrently connected to the same instance, the data in the instance SGA is shared among the users. Consequently, the SGA is sometimes referred to as the "shared global area."

system operational attribute

An attribute holding information that pertains to the operation of the directory itself. Some operational information is specified by the directory to control the server, for example, the time stamp for an entry. Other operational information, such as access information, is defined by administrators and is used by the directory program in its processing.

TLS

See **Transport Layer Security (TLS)**

think time

The time the user is not engaged in actual use of the processor.

throughput

The number of requests processed by Oracle Internet Directory per unit of time. This is typically represented as "operations per second."

Transport Layer Security (TLS)

A protocol providing communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that prevents eavesdropping, tampering, or message forgery.

trusted certificate

A third party identity that is qualified with a level of trust. The trust is used when an identity is being validated as the entity it claims to be. Typically, the certificate authorities you trust issue user certificates.

trustpoint

See **trusted certificate**.

UCS-2

Fixed-width 16-bit **Unicode**. Each character occupies 16 bits of storage. The Latin-1 characters are the first 256 code points in this standard, so it can be viewed as a 16-bit extension of Latin-1.

Unicode

A type of universal character set, a collection of 64K characters encoded in a 16-bit space. It encodes nearly every character in just about every existing character set standard, covering most written scripts used in the world. It is owned and defined by Unicode Inc. Unicode is canonical encoding which means its value can be passed around in different locales. But it does not guarantee a round-trip conversion between it and every Oracle character set without information loss.

UNIX Crypt

The UNIX encryption algorithm.

UTC (Coordinated Universal Time)

The standard time common to every place in the world. Formerly and still widely called Greenwich Mean Time (GMT) and also World Time, UTC nominally reflects the mean solar time along the Earth's prime meridian. UTC is indicated by a z at the end of the value, for example, 200011281010z.

UTF-8

A variable-width encoding of **UCS-2** which uses sequences of 1, 2, or 3 bytes per character. Characters from 0-127 (the 7-bit ASCII characters) are encoded with one byte, characters from 128-2047 require two bytes, and characters from 2048-65535 require three bytes. The Oracle character set name for this is UTF-8 (for the Unicode 2.1 standard). The standard has left room for expansion to support the UCS4 characters with sequences of 4, 5, and 6 bytes per character.

wallet

An abstraction used to store and manage security credentials for an individual entity. It implements the storage and retrieval of credentials for use with various cryptographic services. A wallet resource locator (WRL) provides all the necessary information to locate the wallet.

wait time

The time between the submission of the request and initiation of the response.

X.509

A popular format from ISO used to sign public keys.

Index