Oracle® Internet Directory

Administrator’s Guide

Release 2.0.6

December 1999

Part No. A77230-01

ORACLE®

Oracle Internet Directory Administrator's Guide, Release 2.0.6

Part No. A77230-01

Primary Author: Richard Smith

Contributing Authors: Tridip Bhattachrya, Gilbert Gonzalez, Ashish Kolli, Michelle Sedlak, Elna Tynes, Uppili Srinivasan, Sandy Venning

Contributors: Margaret Chou, Raj Gupta, Brian Gustafsen, Stephen Lee, Harpinder Madan, Michael Mesaros, Radikah Moolky, Howard Mullings, Olaf Stullich, Roger Raj, David Reid, David Saslav, Hari Sastry, Gurudat Shakshikumar, Amit Sharma, Daniel Shih, Saurabh Shrivastava, Deborah Steiner

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 disclaims liability for any damages caused by such use of the Programs.

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.

Contents

2 Concepts and Architecture

3 Preliminary Tasks

4 Using the Administration Tools

Part II Managing Oracle Internet Directory

5 Managing an Oracle Directory Server

6 Managing Directory Schema

7 Managing Directory Entries

8 Managing Secure Sockets Layer (SSL)

9 Managing Directory Access Control

10 Managing Directory Replication

11 Managing National Language Support (NLS)

14 High Availability And Failover

Part IV Appendixes

A Syntax for LDIF and Command Line Tools

B Adding a DSA Using the Database Copy Procedure

C Troubleshooting

D Using Oracle Wallet Manager

E Using Access Control Directive Format

Send Us Your Comments

Oracle Internet Directory Administrator’s Guide, Release 2.0.6

Part No. A77230-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. 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 about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments and suggestions about this manual to the Information Development department at the following e-mail address:

- E-mail - infodev@us.oracle.com
- FAX - (650) 506-7228. Attn: Oracle Internet Directory Documentation Manager
- Postal service:
  Oracle Corporation
  Oracle Internet Directory Documentation Manager
  500 Oracle Parkway, 4op7
  Redwood Shores, CA 94065
  U.S.A.

If you would like a reply, please give your name, address, and telephone number below.

____

____

____

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

Preface

The *Oracle Internet Directory Administrator's Guide* describes the features, architecture, and administration of Oracle Internet Directory. For information about installation, see the Oracle 8*i* Release 8.1.6 installation documentation for your platform.

Intended Audience

This manual is intended for anyone who performs system 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 platform-independent.

How This Book Is Organized

This book contains the chapters and appendixes listed below. Oracle Corporation encourages you to read the conceptual and other introductory material presented in Part I before attempting 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, instructions for starting a directory server, and an introduction to the various administration tools. Specific chapters and their descriptions are:

Chapter 1 Provides an introductory overview of directories, LDAP, and Oracle Internet Directory features.

Chapter 2 Gives an overview of online directories and Lightweight Directory Access Protocol (LDAP). Provides conceptual descriptions of directory entries, object classes, schemas, distributed directories, security, and National Language Support. This chapter also discusses Oracle Internet Directory architecture, and provides suggestions for further reading.

Chapter 3 Discusses how to prepare your directory for running Oracle Internet Directory tools, loading data, and making directory configurations. It tells you how to start and stop OID Monitor and instances of Oracle Directory Server and Oracle Directory Replication Server. It also discusses the need to reset the default security configuration installed with Oracle Internet Directory.

Chapter 4 Explains how to use the various Oracle Internet Directory administration tools: Oracle Directory Manager, command line tools, bulk tools, Catalog Management tool, and OID Password Utility

Part II: Managing Oracle Internet Directory

Part II guides you through the tasks required to configure and maintain Oracle Internet Directory. Specific chapters and their descriptions are:

Chapter 5 Provides instructions for running the Oracle Internet Directory processes, including connect instructions and detailed charts of configuration parameters

Chapter 6 Explains how to administer the Oracle Internet Directory schema

Chapter 7 Explains how to view, add, and modify entries

Chapter 8 Introduces and explains how to configure the features of Secure Sockets Layer (SSL)

Chapter 9 Explains how to control access to the directory

Chapter 10 Explains replication—installing it the first time, and installing new nodes into an environment with replication already installed

Chapter 11 Discusses National Language Support (NLS) as used by Oracle Internet Directory

Part III: Deploying Oracle Internet Directory

Part III discusses deployment considerations. Specific chapters and their descriptions are:

Chapter 12 Gives guidelines for capacity planning an Oracle Internet Directory installation

Chapter 13 Gives guidelines for tuning an Oracle Internet Directory installation

Chapter 14 Discusses the high availability and failover features and deployment guidelines for Oracle Internet Directory

Part IV: Appendixes

Appendix A Provides syntax, usage notes, and examples for LDAP Data Interchange Format and LDAP command line tools

Appendix B Describes an alternate method of adding a node to a replicated directory system if the directory is very large

Appendix C Lists possible failures and error codes and their probable causes

Appendix D Describes and explains how to use Oracle Wallet Manager

Appendix E Describes the format (syntax) of Access Control Information Items(ACIs)

Appendix F Lists schema elements supported in Oracle Internet Directory

Related Oracle Documents

For related information, refer to the following:

- Online help available through Oracle Directory Manager
- Oracle8*i* documentation set

> **More Information:** For non-Oracle material about LDAP, see "Further Reading" on page 2-33.

Conventions

The following conventions are used in this manual:

Convention	Meaning
. . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
. . .	Horizontal ellipsis points in statements or commands mean that parts of the statement or command have been omitted.
bold	Boldface text indicates a term defined in the glossary, text you must type in a command, or a subheading.
italics	Italics indicate: ■ In a code example, a variable for which you must supply a value ■ In regular text, special emphasis ■ Book titles
`courier`	Courier is used for user input and code examples.
syntax	This typeface is used for syntax explanations in code examples.
< >	In code examples, angle brackets may enclose user-supplied names.
[ ]	Brackets enclose a choice of optional items from which you can choose one or none.
{ }	Braces enclose a choice of required items from which you can choose one.

Part I

Getting Started

Part I provides an overview of Oracle Internet Directory and its features, a conceptual foundation necessary to correctly configure and manage a directory, and an introduction to the online administration tool, called Oracle Directory Manager.

Specific chapters in this part are:

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

1

Introduction

The following sections discuss online directories, provide an overview of the Lightweight Directory Application Protocol (LDAP), and explains some of the unique features and benefits of Oracle Internet Directory.

- What is a Directory?
- What is LDAP?
- Oracle Internet Directory Features

> **See Also:** Chapter 2 for a fuller explanation of Oracle Internet Directory concepts, components, and architecture

What is a Directory?

A directory is a way of organizing information so that you can find it easily. It lists objects—for example, people, books in a library, merchandise in a department store—and gives details about each one. A telephone book is a familiar type of directory, a card catalog in a library is another, and a department store catalog still another.

In a computerized environment, a directory is a specialized database that stores collections of information about objects. The information in such a directory might represent any resources that require management—for example, employee names, titles, and security credentials, information about e-commerce partners, or information about shared network resources such as conference rooms and printers.

Although a directory is a database, it is designed very differently from a relational database.

- *Directories are primarily read-focused***.** Typical use of a directory involves a relatively small number of data updates, and a potentially very large number of data retrievals. By contrast, a relational database is primarily write-focused, involving continuous recording of transactions, with retrievals done relatively infrequently—when, for example, an employee needs to generate a monthly report.
- *Directories handle relatively simple transactions on relatively small units of data***.** For example, an application might use a directory to store and retrieve an e-mail address, a telephone number, or a digital portrait. By contrast, a relational database is designed to handle large and diverse transactions using large data items and many operations.
- *Directories are 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 have the information locally, then the server must either retrieve it or point the client application to it transparently. By contrast, relational database applications generally expect data to be location-specific, that is, to reside on a particular database server.

What is LDAP?

LDAP (Lightweight Directory Access Protocol) is the emerging Internet standard for directory services. It is based on the earlier ISO X.500 Directory Access Protocol (DAP) standard, but simplifies that standard considerably, allowing LDAP to be more efficient, straightforward, and easier to implement. LDAP is especially suited for deployment with Internet-centric, "thin-client" applications.

Oracle Internet Directory implements Version 3 of LDAP, which 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 a number of important areas:

- Internationalization: LDAP Version 3 allows servers and clients to support characters used in every language in the world.
- Referrals (also called knowledge references): 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 partition a **Directory Information Tree (DIT)** (described in Chapter 2) across multiple LDAP servers, enabling global deployment.
- Security: A standard mechanism for supporting Simple Authentication and Security Layer (SASL) and Transport Layer Security (TLS) were added, providing LDAP with 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 what LDAP protocols are supported and a description of the directory schema.

See Also:

- RFCs 2251-2256 of the Internet Engineering Task Force, available on the Worldwide Web at:

  http://www.ietf.org/rfc.html

- "Further Reading" on page 2-33 for an additional list of resources on LDAP

Oracle Internet Directory Features

Oracle Internet Directory is a directory service implemented as an application on the Oracle 8*i* database. It enables retrieval of information about dispersed users and network resources. It combines Lightweight Directory Access Protocol (LDAP) Version 3, the open Internet standard directory access protocol, with the high performance, scalability, robustness, and availability of the Oracle8*i* Server.

Oracle Internet Directory includes:

- Oracle Directory Server, which responds to client requests for information about people and resources, and to updates of that information, using a multi-tiered architecture directly over TCP/IP
- Oracle Directory Replication Server, which replicates LDAP data between Oracle Directory Servers
- Oracle Directory Manager, a graphical user interface administration tool
- A variety of command line administration and data management tools

Oracle Internet Directory provides three solid benefits as described in the following subsections:

- Scalability
- High Availability
- Security

Scalability

Oracle Internet Directory exploits the massive strengths of the Oracle8*i* database, enabling support for terabytes of directory information. In addition, technologies such 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 mission critical deployment applications. One way this is reflected is in its replication capability. Oracle Internet Directory supports full, multi-master replication between directory servers. This means that if one server in a replication community is unavailable for any reason, a user can access the directory data from another server. Information about changes made to data on a server is stored in special tables on the Oracle8*i* database. These are replicated throughout the directory environment by Oracle's Advanced Symmetric Replication (ASR), a robust, field-proven replication mechanism.

Oracle Internet Directory also leverages all of the availability features of the Oracle8*i* database server. Because directory information is stored securely in the Oracle8*i* database, it is protected by Oracle's backup capabilities. Additionally, the Oracle8*i* database, running with large datastores and heavy loads, can recover from system failures quickly.

Security

Oracle Internet Directory offers comprehensive and flexible support for directory access control. This includes entry level, attribute level, and prescriptive access control to provide varying levels of security to meet the specific needs of enterprise and service providers. An administrator can grant or control access to a specific directory object or to an entire directory subtree. Oracle Internet Directory implements three levels of user authentication: anonymous, password-based, and certificate-based using Secure Sockets Layer (SSL) Version 3 for authenticated access and data privacy.

2

Concepts and Architecture

This chapter explains some of the key concepts you need to know when using Oracle Internet Directory. It covers topics in the following sections:

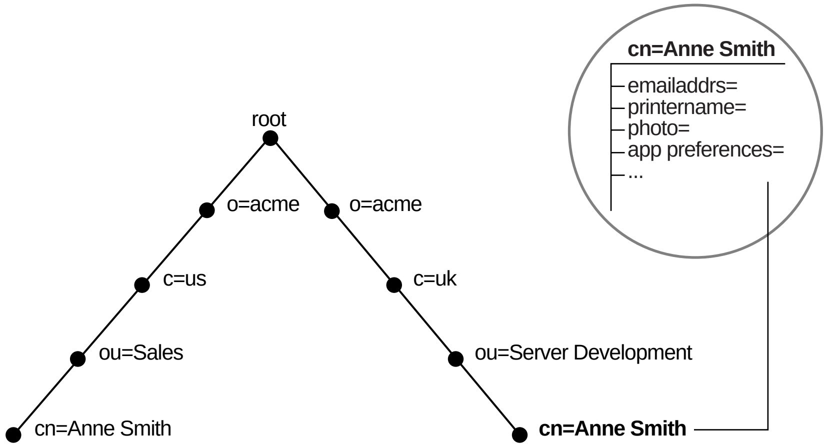
Entries

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 is structured along geographical and organizational lines. It diagrammatically represents entries for two employees who have the same common name (cn), namely, Anne Smith.

The DIT branch on the right represents the entry for the Anne Smith whose common name (cn) is Anne Smith, who works in the organization (o) Acme, in the country (c) of Great Britain (uk), in an organizational unit (ou) named Server Development.

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 entry's full DN—not simply it's RDN. For example, within the global organization in Figure 2–1, there are two employees with the same RDN, namely, `Anne Smith`. To avoid confusion between these two entries, 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.)

Attributes

In a typical telephone directory, an entry for a person contains such information items as an address and phone number. In an online directory, such information items are called *attributes*. Attributes in a typical employee entry could include a job title, e-mail address, and phone number.

In Figure 2–1, the entry for Anne Smith in Great Britain 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 2–1 is also an entry with attributes, although the attributes for each are not shown.

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

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 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.

Single-Valued and Multi-Valued Attributes

Single-valued attributes carry only one value in the attribute, whereas *multi-valued* attributes can have more than one value. An example of a multi-valued attribute is a group membership list that includes the names of all the members of a given group, such as an e-mail list. Attributes can be either single-valued or multi-valued.

See Also:

- "Adding Group Entries" on page 7-18
- "ldapmodify" on page A-11

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	Example of Attribute Value
commonName	`cn`	Common name of an entry	`cn=Anne Smith`
domainComponent	`dc`	Component in a Domain Name System (DNS)	`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	`Acme`
organizationalUnitName	`ou`	Name of a unit within an organization	`ou=Server Development`
owner	`owner`	Distinguished name of the person who owns the entry	The following is a line in an LDIF file: `owner: cn=Anne Smith, ou=Server Development, o= Acme, c=uk`
surname, sn	`sn`	Last name of a person	`Smith`

telephoneNumber	`telephoneNumber`	Telephone number	`telephoneNumber=(650) 123-4567`
			or
			`telephoneNumber=6501234567`

In addition, Oracle Internet Directory provides several proprietary attributes that are listed in Appendix F.

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 can be printed, whether it has to be in the form of a date, or whether the data can consist of numbers only. Each attribute can have only one syntax attached to it.

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

See Also: "LDAP Syntax" on page F-6

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 relevant *matching rules* 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 F-9

Object Classes

When you define a directory entry, you assign one or more *object classes* to it. These object classes contain *attributes*. An object class is a category of objects, and it typically provides both *mandatory* and *optional* attributes for particular objects.

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

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

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

The remainder of this section discusses object classes in the following subsections:

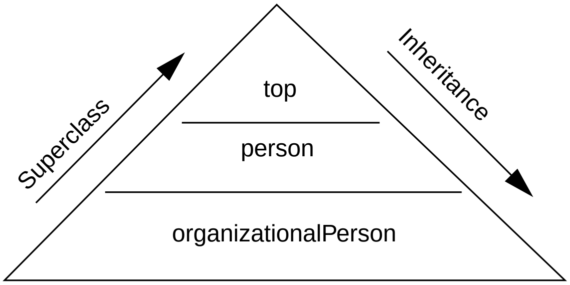
- Subclasses, Superclasses, and Inheritance
- Object Class Types

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

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`.

Figure 2–2 Object Class Inheritance



A subclass **inherits** all of the attributes belonging to its superclass. For example, in Figure 2–2, the `organizationalPerson` object class inherits the attributes of the `person` object class. 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 contains values. When a subclass inherits attributes from a superclass, it inherits only the attribute framework—not the attribute *values*—of the superclass

As you can also see in Figure 2–2, one special object class, called `top`, has no superclasses. It is one of the superclasses of every structural object class (discussed in "Object Class Types" on page 2-9) in the directory, and its attributes are inherited by every entry.

Object Class Types

There are three types of object classes described in the following sections:

- Abstract Object Classes
- Structural Object Classes
- Auxiliary Object Classes

Abstract Object Classes

These object classes are virtual object classes. An abstract object class cannot be the only object class for an entry. The object class `top` is an abstract object class. It is required to be 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 following list contains the names of the optional object classes in `top`, and either describes or points to further information about each one.

- orclGuid—Global identification which remains constant if the entry is moved
- creatorsName—See the appropriate IETF documentation.
- createTimestamp—See the appropriate IETF documentation.
- orclACI—See "orclACI" on page 9-2
- orclEntryLevelACI—See "orclEntryLevelACI" on page 9-3.

Structural Object Classes

Most of the object classes that you use are structural object classes, and all entries should belong to one structural object class. Examples of structural object classes are `person` and `groupOfNames.` These object classes follow DIT structure rules that specify which kinds of object classes can be created under any given object class. For example, in Figure 2–1 on page 2-2, a DIT structure rule might require that all objects that are located below the `organization` (o) Acme must be `organizational units` (ou). Following this rule, you could not enter `person` objects directly below the `organization` Acme.

Auxiliary Object Classes

These object classes are groupings of attributes that can be used to expand the existing list of attributes in an entry. For example, suppose you have defined an entry as a member of two object classes. If you want that entry to have additional attributes that do not belong to these object classes, you can create a new *auxiliary* object class that contains the extra attributes, and associate it 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.

Schemas

All information about how data is organized in the DIT—that is, metadata such as object classes, attributes, matching rules, and syntaxes—is included in what is called the directory schema. This information is stored 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: Appendix F for a list of both standard and proprietary schema elements installed with Oracle Internet Directory

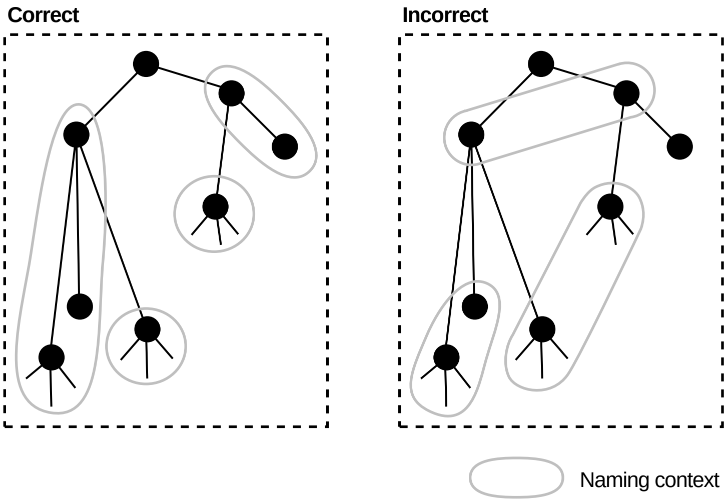
Distributed Directories

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

When physically distributing information, a directory divides its information into units called directory *naming contexts*. A directory 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.

Figure 2–3 Valid and Invalid Naming Contexts



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.

The following pages discuss both types of directory distribution in the following sections:

- Replication
- Partitioning

Replication

This section describes topics in the following subsections:

- Overview of Replication
- Directory Replication Groups and Replication Agreements
- Oracle Advanced Symmetric Replication (ASR)
- Replication Architecture
- How Replication Works

Overview of Replication

In addition to improving performance by providing more servers to handle queries, replication also improves reliability by eliminating risks associated with a single point of failure.

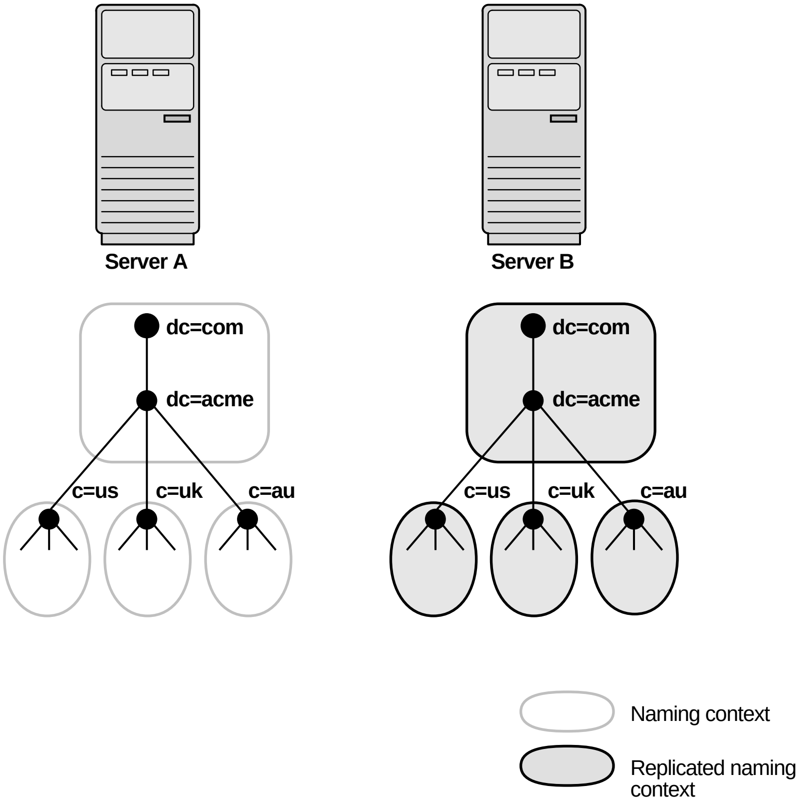
Note: There are no Internet standards for directory replication yet, though such standards are being developed by the IETF. Oracle Internet Directory replication adheres to the IETF standard proposal for representing directory change information in **change logs**.

For more on change logs, see "Replication Architecture" on page 2-15.

Each copy of a naming context that is contained within a server is called a *replica*. A directory server can hold read-only and updatable replicas. Servers that hold updatable replicas are called *suppliers*. Their changes are propagated to other servers called *consumers*. You specify how many times the replication server should try to apply changes to consumers. Once that number is reached, the replication server moves the changes to a queue, then attempts to apply them at regular, less frequent intervals that you specify.

Figure 2–4 shows a replicated directory.

Figure 2–4 A Replicated Directory



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.

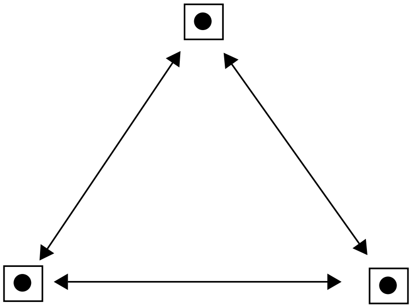
Directory Replication Groups and Replication Agreements

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 multi-master replication.

Figure 2–5 illustrates a Directory Replication Group in which three nodes share updates with each other in a replication agreement.

Figure 2–5 Directory Replication Group



In Figure 2–5, 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:** "Step 6: Configure Replication" on page 10-9 for information about how to configure replication agreements

Oracle Advanced Symmetric Replication (ASR)

Transport of update information between nodes is managed by Oracle Advanced Symmetric Replication (ASR), a store-and-forward transport feature available in Oracle8*i*. It allows database tables to be kept synchronized across two Oracle databases. Directory replication uses this ASR mechanism as a transport vehicle for propagating directory change information across to other directory servers participating in a replication agreement. ASR 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.

ASR environments allow read and update access to directory tables anywhere in the system. Typical ASR configurations use row-level replication with asynchronous data propagation.

ASR 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:

- Chapter 10 for more information on replication
- *Oracle8i Replication* for information about ASR

Replication Architecture

Replication uses change logs on supplier servers. 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.

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 LDAP-compliant directory server which implements them. The replication process is illustrated in Figure 2–6 on page 2-17.

To configure replication, you specify which nodes in a replication group should share changes.

See Also: "Step 6: Configure Replication" on page 10-9 for information on configuring replication

How Replication Works

This section provides a general overview of the replication process on both the supplier and the consumer.

On the supplier side:

1. The LDAP Server generates a change log object in the change log object store when it performs any directory modification that was issued by an LDAP client.
2. The replication server spawns an outbound change log processing thread based on a scheduled replication cycle to translate the change log object into a row—for example, Change entry—in the change log table.
3. When a change entry gets inserted and then committed into the change log table, ASR immediately copies the change into the deferred transaction queue.
4. After a scheduled ASR replication interval, ASR pushes pending transactions from the deferred transaction queue across the network to the consumer change log table.

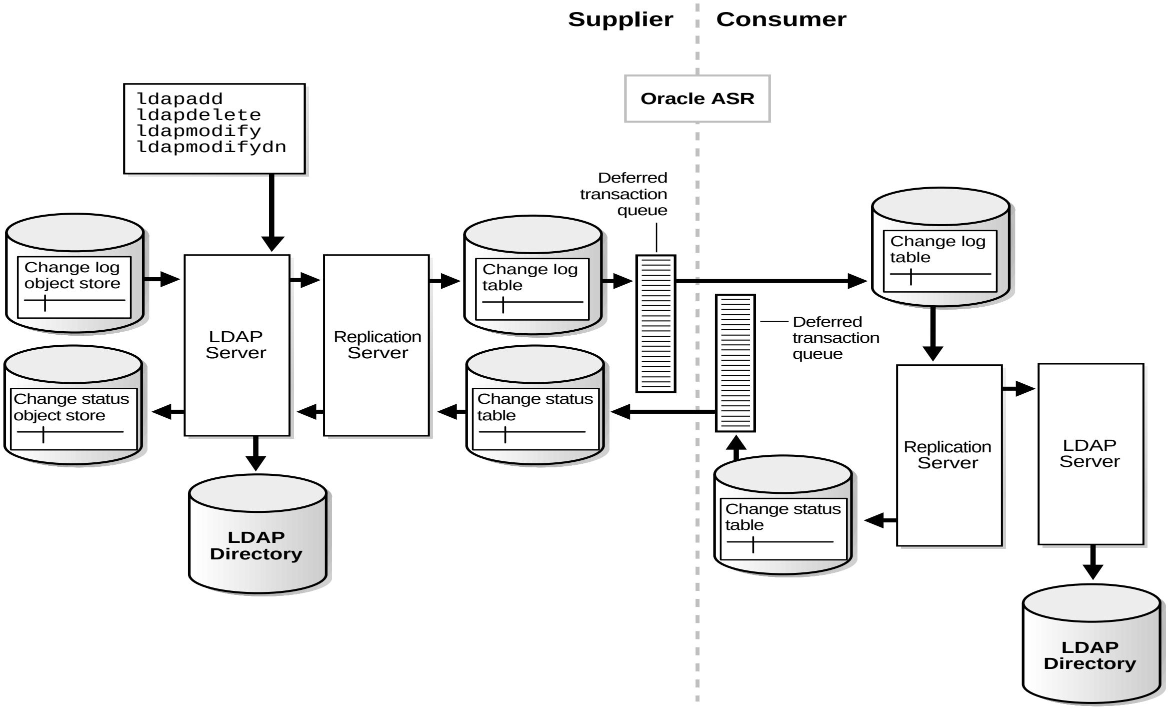
On the consumer side:

1. The replication server spawns a change log processing thread for each supplier, based on a scheduled replication cycle.
2. The change log processing thread consults the change status table for the last change applied from the supplier to the consumer directory.
3. It then fetches and applies all the new changes from the change log table to the LDAP Server.
4. After completing the change log processing, the change log processing thread updates the change status table to record the last change applied from the supplier before exiting.
5. ASR copies the change status update into the deferred transaction queue.
6. After the scheduled ASR replication interval, ASR pushes pending change status updates from the deferred transaction queue to the supplier change status table.

> **See Also:** "The Replication Process" on page 10-27 for a more detailed explanation of how the replication server adds, deletes, and modifies entries, as well as how it modifies DNs and RDNs

Figure 2–6 illustrates the replication process.

Figure 2–6 Overview of ASR-Based Replication Architecture



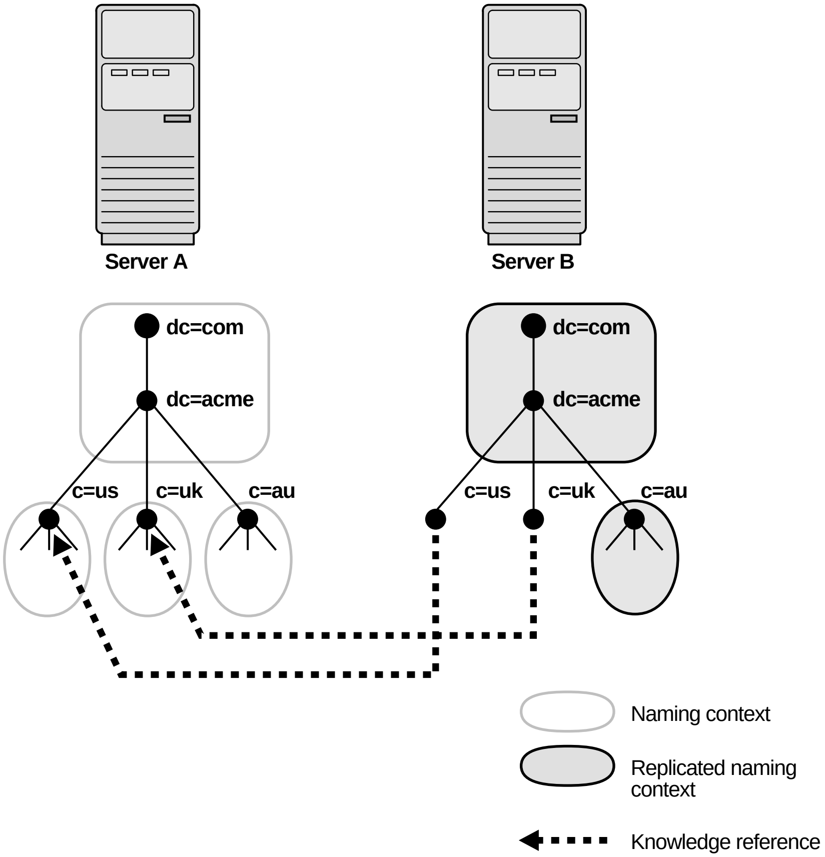
Although, in Figure 2–6, the roles of supplier and consumer have been separated, in an actual multi-master replication environment, every directory server is both a supplier and a consumer. In such an environment, *garbage collection*—that is, the removal of entries that are already applied or that have been dropped as candidate changes—occurs regularly. *Remote* change records in the local Changelog table are deleted by the garbage collection thread if they have been applied locally, and *local* change records in the local Changelog table are deleted by the garbage collection thread if they have been distributed to all the consumers.

> **See Also:** "Conflict Resolution" on page 10-23 for a detailed description of how the replication server resolves conflict when it adds, deletes, and modifies entries and when it modifies DNs and RDNs

Partitioning

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`

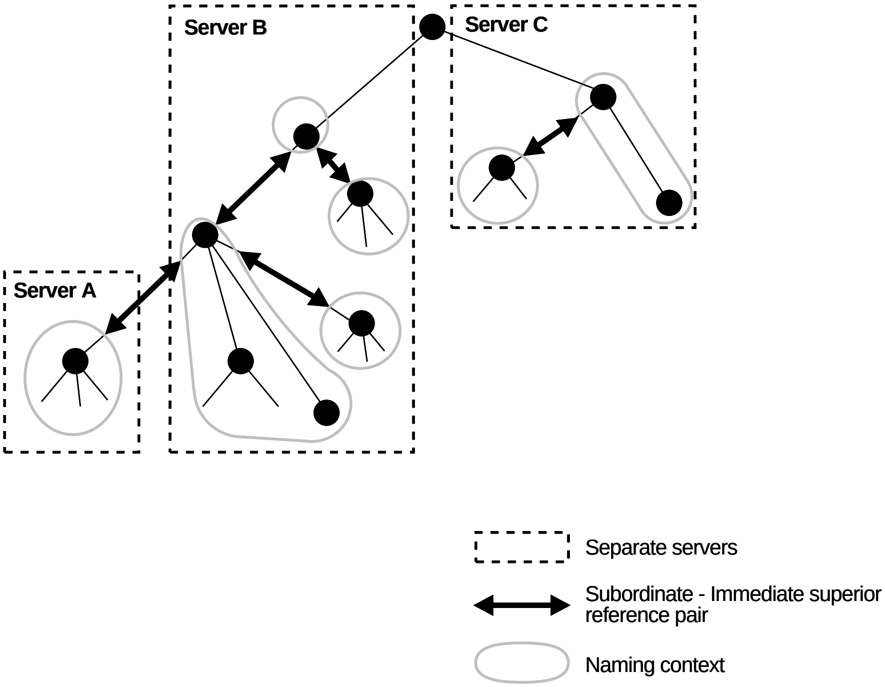
Two naming contexts on Server A are replicated on Server B:

- `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. Knowledge references provide the names and addresses of the various naming contexts. Server B uses knowledge references to tell clients that Server A has the requested information—in this case, 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.

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



For example, in Figure 2–8, two of the naming contexts on Server B have naming contexts that are subordinate to them. 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.

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 the 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.

Security

Oracle Internet Directory provides many powerful mechanisms for protecting information from unauthorized access. This section discusses the following elements of directory security:

- Authentication: Ensuring that users', hosts', and clients' identities are correctly validated
- Access Control and Authorization: Ensuring that a user reads or updates only the information for which that user has privileges
- Data Integrity: Ensuring that data is not modified during transmission
- Data Privacy: Ensuring that data is not disclosed during transmission

Authentication

Authentication is the process by which the directory server establishes the true identity of the user connecting to the directory. Authentication occurs when an LDAP session is established by means of the ldap-bind operation. Thus every session has an associated user identity. This identity is also referred to as an authorization ID.

To ensure that users', hosts', and clients' identities are correctly known, the Oracle Internet Directory provides the three authentication options described in the following sections:

- Anonymous Authentication
- Simple Authentication
- Strong Authentication with SSL

Anonymous Authentication

If your directory is available to everyone, then you can allow users to log in to the directory anonymously. In this case, users simply leave blank the user name and password fields when they log in. Each anonymous user then exercises whatever privileges are specified for anonymous users.

Simple Authentication

In this case, 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 match the DN and password stored in the directory.

Strong Authentication with SSL

Oracle Internet Directory provides strong authentication by using public-key encryption available with Secure Sockets Layer (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.

The remainder of this section discusses SSL in the following subsections:

- About SSL
- Components of SSL
- The SSL Handshake
- SSL and Oracle Internet Directory

About SSL SSL is an industry standard protocol for securing network connections. SSL provides authentication through the exchange of certificates that are verified by trusted certificate authorities. A certificate ensures that an entity’s identity information is correct.

You can use SSL in one of three authentication modes. You can require:

SSL Mode	Description
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.
One-way authentication	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
Two-way authentication	Both client and server authenticate themselves to each other. Both the client and server send certificates to each other.

Components of SSL The components of SSL include:

- **Certificate**

  A certificate ensures that the entity’s identity information is correct and that the public key actually belongs to that entity. A certificate is created when an entity’s public key is signed by a trusted identity, that is, a certificate authority (CA), described more fully in this section. A certificate contains the entity’s name, public key, serial number, and expiration date. It may contain information about the privileges associated with the certificate. Finally, it contains information about the CA that issued it. A certificate is valid until it expires or is revoked.

- **Certificate Authority (CA)**

  A certificate authority is 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.

Different CAs may have different identification requirements when issuing certificates. One certificate authority may want to see a user's driver's license, another may want the certificate request form to be notarized, yet another may want fingerprints of the person requesting a certificate. The certificate authority publishes its own certificate which includes its public key.

Each network entity has a list of the certificates of the CAs it trusts. Before communicating with another entity, a given entity uses this list to verify that the signature on the other entity's certificate is from a trusted CA. Network entities can obtain their certificates from the same or from different CAs.

- **Wallet**

  A wallet is an abstraction used to store and manage authentication data such as keys, certificates, and **trusted certificates**, also called trustpoints, which are needed by SSL. In an Oracle environment, each system using SSL has a wallet with an X.509 version 3 certificate, private key, and list of trusted certificates.

  Security administrators use the **Oracle Wallet Manager** to manage security credentials on the server. Wallet owners use it to manage security credentials on clients. Specifically, the Oracle Wallet Manager is used to do the following:

  - Generate a **public/private key pair** and create a certificate request for submission to a certificate authority
  - Install a certificate for the identity
  - Configure trusted certificates for the identity

  See Also: Appendix D for detailed information about managing wallets by using the Oracle Wallet Manager

The SSL Handshake At the beginning of their communication, the client and directory server perform a handshake which includes three important tasks:

- The client and server establish which cipher suite to use. 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 server sends its certificate to the client. The client verifies that the directory server's certificate was signed by a trusted CA.

  Similarly, if client authentication is required, the client sends its own certificate to the directory server. The directory server verifies that the client's certificate was signed by a trusted CA.

- The client and directory server exchange key material using public key cryptography, and, from this material, they each generate a session key. All subsequent communication between client and directory server is encrypted and decrypted by using this set of session keys and the negotiated cipher suite.

See Also:

- "Data Integrity: Ensuring that data is not modified during transmission" on page 2-20 for more information on data integrity and encryption
- "Supported Cipher Suites" on page 8-2 for a list of SSL cipher suites supported in Oracle Internet Directory

SSL and Oracle Internet Directory SSL authentication between a client and a directory server involves three basic steps:

1. The user initiates an LDAP connection to the directory server by using SSL on the SSL port. (The default SSL port is 636.)
2. SSL performs the handshake between client and directory server.
3. If the handshake is successful, the directory server verifies that the user has the appropriate authorization to access the directory.

See Also: Chapter 9 for information on specifying authentication

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 ID 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 (ACI) is the directory metadata that captures the administrative policies relating to access control.

ACI is stored in Oracle Internet Directory as user-modifiable operational attributes. 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.

ACIs are represented and stored as text strings in the directory. These strings must conform to a well defined format. Each valid value of an ACI attribute represents a distinct access control policy. These individual policy components are referred to as ACI Directives or ACI Items and their format is called the ACI Directive format.

See Also:

- Chapter 9 for instructions on setting access control policies
- Appendix E for instructions on correctly formatting ACI directives

Data Integrity

To ensure that data has not been modified, deleted, or replayed during transmission, Oracle Internet Directory requires the use of 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

To ensure that data is not disclosed during transmission, the 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 8 for more information about SSL

National Language 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 that Oracle NLS supports. Moreover, although several different APIs are involved in the Oracle Internet Directory implementation, Oracle Internet Directory ensures that the correct character encoding is used with each API.

NLS uses both single-byte and multi-byte characters. A single-byte character is represented by one byte of memory. ASCII text, for example, uses single-byte characters. By contrast, a multi-byte character can be represented by more than one byte. Simplified Chinese, for example, uses multi-byte 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), and Oracle Directory Replication Server (OIDREPLD)—always use the UTF-8 character set by default.

Oracle Directory Manager, a Java-based tool, internally uses Unicode (UCS2—that is, fixed-width 16-bit Unicode). In Java, UCS2 is the easiest way to handle characters—including English characters. The Java client uses standard Java packages to convert both to and from UCS2 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 2-27 for information on the main Oracle Internet Directory components
- Chapter 11 and *Oracle8i National Language Support Guide* for a detailed discussion of NLS

Oracle Internet Directory Architecture

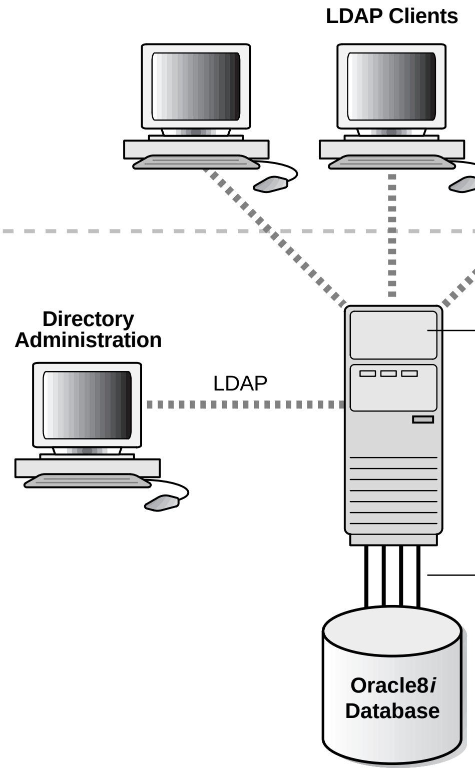
This section covers the topics in the following sections:

- Architectural Overview
- An Oracle Internet Directory Node
- An LDAP Directory Server Instance
- Configuration Set Entries
- How Oracle Internet Directory Works: An Example

Architectural Overview

An Oracle Internet Directory node is implemented as an application running on the Oracle 8*i* server. To communicate with the database, which may be on the same platform or on a different one, the Oracle Internet Directory uses **Net8**, Oracle's platform-independent database connectivity solution. This relationship is illustrated in Figure 2–9.

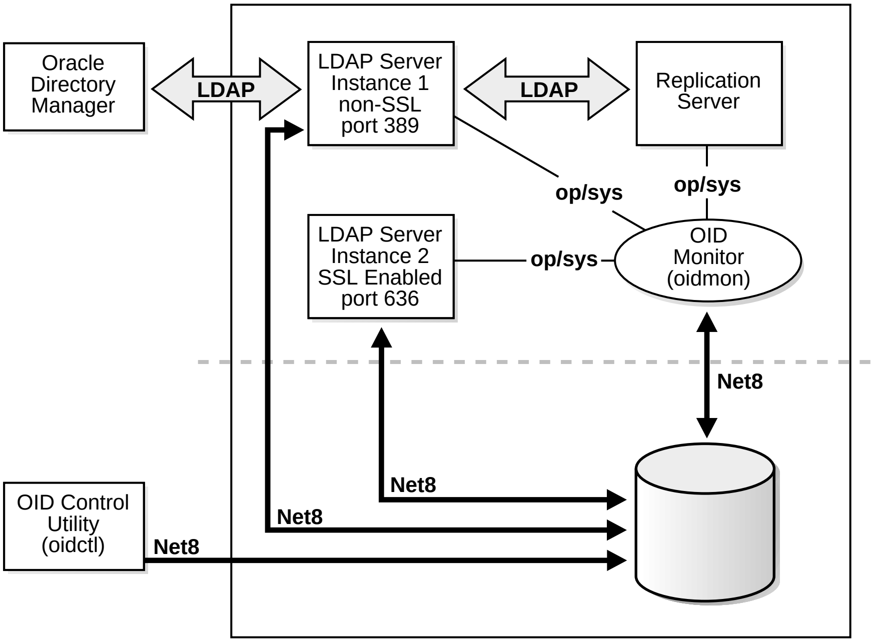
Figure 2–9 Oracle Internet Directory Architecture



An Oracle Internet Directory Node

Figure 2–10 shows the various directory server components and their relationships running on a single node.

Figure 2–10 A Typical Oracle Internet Directory Node


Oracle Directory Manager
LDAP
LDAP Server Instance 1 non-SSL port 389
LDAP
Replication Server
op/sys
op/sys
LDAP Server Instance 2 SSL Enabled port 636
op/sys
OID Monitor (oidmon)
Net8
Net8
Net8
OID Control Utility (oidctl)
Net8

Note: In Figure 2–10, the database is on the same node as the directory server processes. However, since all connections with the database are through Oracle Call Interface (OCI) and Net8, it is possible to use a database on a different server.

An Oracle Internet Directory node (Figure 2–10) includes the following major components:

Component	Description
LDAP 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 LDAP server instance on a node, each listening on a different port. Oracle Internet Directory dispatcher and server processes use multiple threads.
Replication server	Tracks and sends changes to replicated servers in an 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. **See Also:** "Distributed Directories" on page 2-10, Chapter 10, and Appendix B
Oracle8*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.
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. **See Also:** "Using OID Control Utility" on page 4-15

OID Control Utility 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 Internet Directory instance.

> **See Also:** "Configuration Set Entries" on page 2-32 for more detailed information about the configuration parameters

OID Monitor executes the start-server and stop-server requests that you initiate from OID Control Utility. OID Monitor is also responsible for monitoring servers and restarting 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 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's file system.

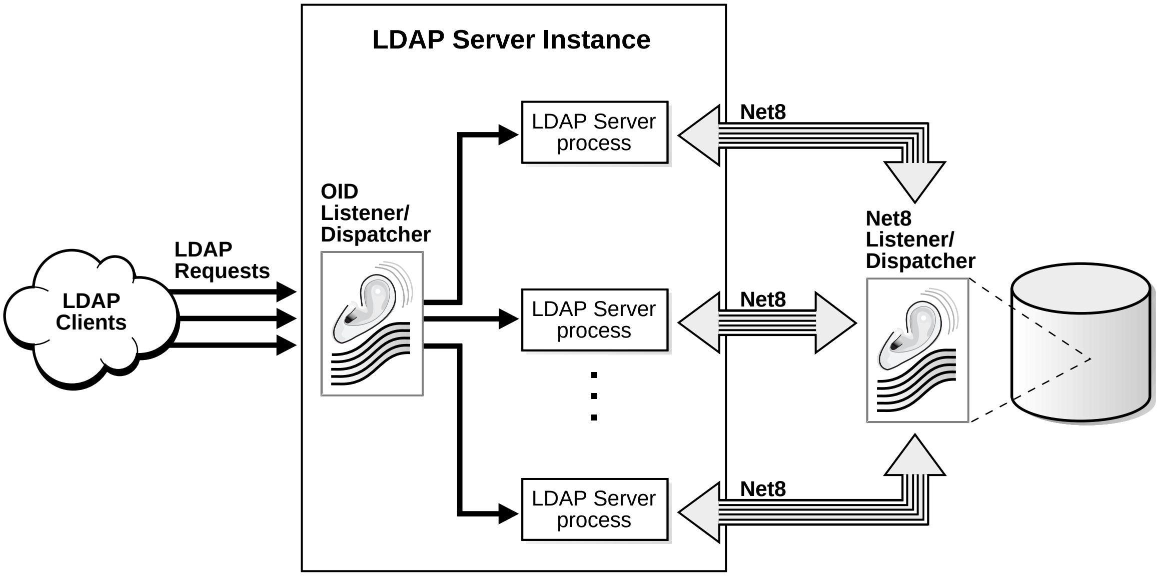
OID Monitor checks the state of the servers through mechanisms provided by the operating system.

The replication server uses LDAP to communicate with an LDAP server instance. To communicate with the database, all components use OCI / Net8. Oracle Directory Manager and the command line tools communicate with the Oracle Internet Directory (LDAP) servers over LDAP.

An LDAP Directory Server Instance

Each LDAP directory server instance looks similar to what Figure 2–11 illustrates.

Figure 2–11 LDAP Server Instance Architecture



The components of the LDAP directory server instance include:

- An Oracle Internet Directory dispatcher process which has the listener / dispatcher mechanism that listens for LDAP commands at its port.
- Server processes, created when the LDAP directory server instance starts. Multiple server processes allow Oracle Internet Directory to take advantage of multiple processor systems. The number created is determined by a configuration parameter (ORCLSERVERPROCS). The default is one. A "worker thread" for each operation processes the client's request.
- Database connections from each server process, which are spawned as needed, up to a maximum number determined by a configuration parameter (ORCLMAXCC). The default value for this parameter is 10.

Configuration Set Entries

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

See Also: Table F–1 on page F-5 for a list of configuration set entry attributes

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 configsets 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 5-2.

How Oracle Internet Directory Works: An Example

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, 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, 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)/Net8 and sends it to the Oracle8*i* database.
6. The Oracle8*i* database fetches the information and passes it back through the chain—to the directory server, then to the C API, and, finally, to the client.

Further Reading

For more information on concepts discussed in this chapter, refer to the list of books and online articles below. Most of the entries in the list also contain references to other publications.

Chadwick, David. *Understanding X.500 The Directory.* Thomson Computer Press, 1996. This book is now out of print, but is available online at: http://www.salford.ac.uk/its024/Version.Web/Contents.htm

Hodges, Jeff, Computing and Communication Services, Stanford University, http://www-leland.stanford.edu/group/networking/directory/x500ldapfaq.html

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.

Kosiur, Dave, LDAP: "The next-generation directory?," *SunWorld Online,* October 1997.

Radicati, Sara, *X.500 Directory Services, Technology and Deployment,* International Thomson Computer Press, 1994.

University of Michigan LDAP Repository, http://www.umich.edu/~rsug/ldap/

3

Preliminary Tasks

Before you can run the administration tools and begin configuring and deploying the directory, you need to start OID Monitor and start a directory server instance. You also need to reset the default security configuration.

This chapter discusses topics in the following sections:

Step One: Start the OID Monitor Daemon

The OID Monitor daemon must be running to process the start-server and stop-server commands that you initiate through the **OID Control Utility**.

This section covers topics in the following subsections:

- Starting the OID Monitor Daemon
- Stopping the OID Monitor Daemon

Starting the OID Monitor Daemon

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
   ```

   See Also: Chapter 11.

2. At the system prompt, type:

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

Argument	Description
net_service_name	Connect string 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.
seconds	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 Daemon

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

```
oidmon connect=net_service_name sleep=seconds stop
```

Argument	Description
net_service_name	Name of the database to which you want to connect. This is the network service name set in the `tnsnames.ora` file.
seconds	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.
stop	Stops the OID Monitor process

For example:

```
oidmon connect=dbsl stop>
```

Step Two: Start Server Instances

Once the OID Monitor is running, start and stop server instances 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 covers topics in the following subsections:

- Starting an LDAP Server Instance
- Stopping an LDAP 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 LDAP Server Instance

The syntax for starting an LDAP server instance is:

```
oidctl connect=net_service_name server=oidldapd instance=server_instance_number
[configset=configset_number] [flags=' -p port_number -debug debug_level -l
change-logging -server n'] start
```

For example, to start an LDAP server instance whose net service name is dbs1, using configset5, at port 12000, with a debug level of 1024, an instance number 3, and turning off change-logging, 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 LDAP 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.

Table 3–2 provides descriptions for each of the arguments for the command line syntax. It also contains cross-references to concepts explained elsewhere in this book and in other Oracle documentation.

Table 3–1 Command Line Syntax

Argument	Description	Information
net_service_name	If you already have a `tnsnames.ora` file configured, this is the name specified in that file, located in ORACLE_HOME/network/admin	*Net8 Administrator's Guide*
server	Type of server to start (valid values are OIDLDAPD and OIDREPLD). This is not case-sensitive.	"Step Two: Start Server Instances" on page 3-3
server_instance_number	Instance number of the server to start. Should be a number between 0 and 1000.	"Managing Server Configuration Set Entries" on page 5-2
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.	"Configuration Set Entries" on page 2-32
-p *port_num*	Specifies a port number during server instance startup. Default port if not set is 389.	"Configuring SSL Parameters" on page 8-2 and "Managing Server Configuration Set Entries" on page 5-2
-debug *debug_level*	Specifies a debug level during LDAP server instance startup	"Setting Debug Logging Levels by Using the OID Control Utility" on page 5-21
-h *host_name*	Specifies the host name on which the server runs.	
-l	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)	Chapter 10
-server *n*	Specifies the number of server processes to start on this port	
start	Starts the server specified in the *server* argument.	"Step Two: Start Server Instances" on page 3-3

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 LDAP 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=’ -h hostname -p port_number
-d debug_level -z transaction_size ’ start
```

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.

Table 3–2 provides descriptions for each of the arguments for the command line syntax. It also contains cross-references to concepts explained elsewhere in this book.

Table 3–2 Command Line Syntax

Argument	Description	Information
net_service_name	If you already have a tnsnames.ora file configured, this is the name specified in the tnsnames.ora file, located in ORACLE_HOME/network/admin	*Net8 Administrator's Guide*
server	Type of server to start (valid values are OIDLDAPD and OIDREPLD). This is not case-sensitive.	"Step Two: Start Server Instances" on page 3-3
server_instance_number	Instance number of the server to start. Should be a number between 0 and 1000.	"Managing Server Configuration Set Entries" on page 5-2
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.	"Configuration Set Entries" on page 2-32 Table F–1 on page F-5 for a list and descriptions of the entire set of attributes that are used to configure an instance of a directory server
-p *port_num*	Specifies a port number during server instance startup. Default port if not set is 389.	"Configuring SSL Parameters" on page 8-2 and "Managing Server Configuration Set Entries" on page 5-2
-d *debug_level*	Specifies a debug level during replication server instance startup	"Setting Debug Logging Levels by Using the OID Control Utility" on page 5-21
-h *host_name*	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)	Chapter 10
-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.	Chapter 10
start	Starts the server specified in the *server* argument.	"Step Two: Start Server Instances" on page 3-3

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
```

See Also:

- "Starting an LDAP Server Instance" on page 3-4
- "Starting an Oracle Directory Replication Server Instance" on page 3-6

Restarting Directory Server Instances

OID Monitor must be running whenever you start or stop 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 if you want the changed value in the configuration set entry to take effect on that server instance. You can either issue the `stop` command followed by the `start` command, or you can use the *restart* command. The restart command both stops and restarts the server instance.

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

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

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

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

If there were more than one instance of the Oracle Directory Server running on that node using `configset3`, then all the instances could be restarted at once 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 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`.

Step Three: Reset the Default Security Configuration

When you first install the Oracle Internet Directory, the default configuration grants to all users complete access to the directory. One of the first things you need to do is 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

To configure security, you use the administration tools described in Chapter 4.

See Also:

- Chapter 9 for a detailed explanation of access control options and instructions for setting up security
- Chapter 4 for instructions on using Oracle Directory Manager and an overview of the command line tools
- Appendix F or syntax and usage notes for the command line tools

4

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 by using it. It also introduces the command line and bulk tools.

This chapter discusses these topics in the following sections:

Using Oracle Directory Manager

This section describes some of the basic features of Oracle Directory Manager. More specific instructions are found in sections explaining how to perform various tasks.

This section discusses the following tasks and features of Oracle Directory Manager:

- 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 first have a directory **server instance** running.

If you do not already have a server instance running, follow the steps in Chapter 3 before attempting to start Oracle Directory Manager.

> **See Also:** "Oracle Internet Directory Architecture" on page 2-27 for a conceptual explanation of Oracle Directory Server instances

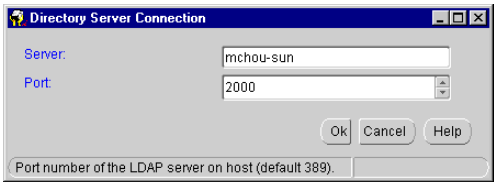
To start Oracle Directory Manager, follow the instructions for your platform:

Platform	Instructions
Windows NT or Windows 95	You can either: ■ Type at the Run command: `oidadmin` ■ Type at a DOS command prompt: `oidadmin`
Sun Solaris	Type at the system prompt: `oidadmin`

When you start Oracle Directory Manager for the first time, an alert tells you that you must connect to a server. Click OK.

Connecting to a Directory Server

1. The Directory Server Connection dialog box prompts you for the name and port number of an available server:



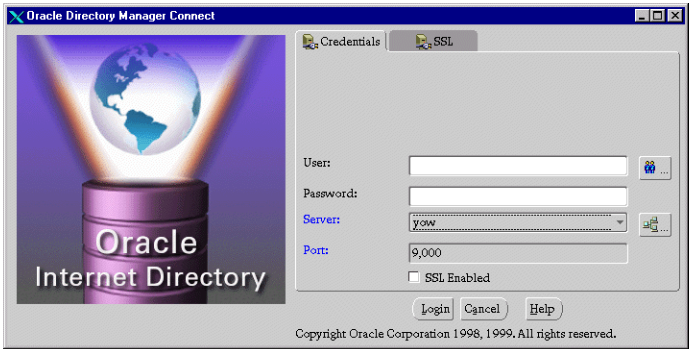
Directory Server Connection

Server: mchou-sun

Port: 2000

Port number of the LDAP server on host (default 389).

The default port is 389. You can change the port if you wish. However, if you have an Oracle Directory Server running on a non-default port, be sure that any clients that will use it are informed of the correct port to use. Click OK. The Oracle Directory Manager Connect dialog box appears:



Oracle Directory Manager Connect

User:

Password:

Server: yow

Port: 9,000

□ SSL Enabled

Copyright Oracle Corporation 1998, 1999. All rights reserved.

2. In each field of the Credentials tab page, type the information specific to this server instance as described in Table 4–1.

Table 4–1 Credential Information

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 or ■ 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`
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`. If you are logging in anonymously, leave the Password box empty. After you are logged into Oracle Directory Manager and have connected to a directory server, you should change this password to protect the directory. For instructions on how to do this, see "Managing Super, Guest, and Proxy Users" on page 5-17.
Server	Select the host containing the directory server to which you want to connect by selecting it from the Server list. If you are already connected to a directory server, and you want to connect to a directory server on a different host, click the button to the right of the Server field. A dialog box presents you with a list of available servers. Select the one you want and click OK. If you want to add a directory server, click Add. The Directory Server Connection dialog box appears. Type the name of the directory server you want to add, then click OK.
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.

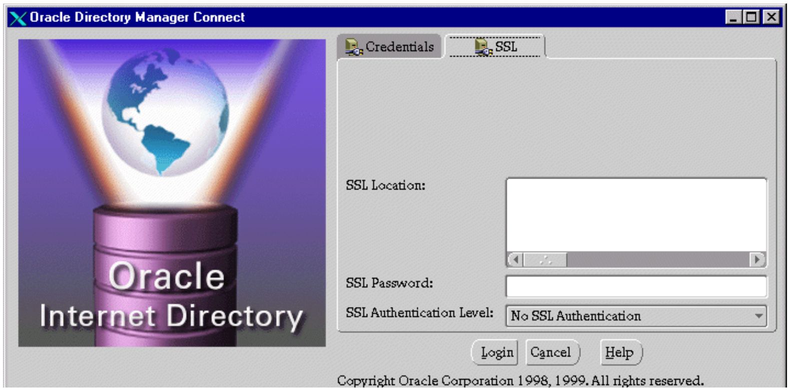
Table 4–1 Credential Information

Field	Description
SSL Enabled	You can connect to a directory server either with or without SSL. If you connect by using Secure Sockets Layer (SSL), then Oracle Directory Manager becomes an SSL client. All commands you issue through Oracle Directory Manager are then sent over SSL. 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, authentication will fail when you try to connect. ■ You have already created a wallet containing a certificate and a list of trusted certificates. Selecting this check box causes all messages you issue by using Oracle Directory Manager to be sent over SSL.

See Also:

- Chapter 8 for instructions on enabling SSL
- Appendix D for instructions on creating a wallet
- "Entries" on page 2-2 for instructions on formatting distinguished names
- "Configuring SSL Parameters" on page 8-2 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:



SSL Authentication Level: No SSL Authentication

Copyright Oracle Corporation 1998, 1999. All rights reserved.

4. Enter the requested data in the fields as described in Table 4–2.

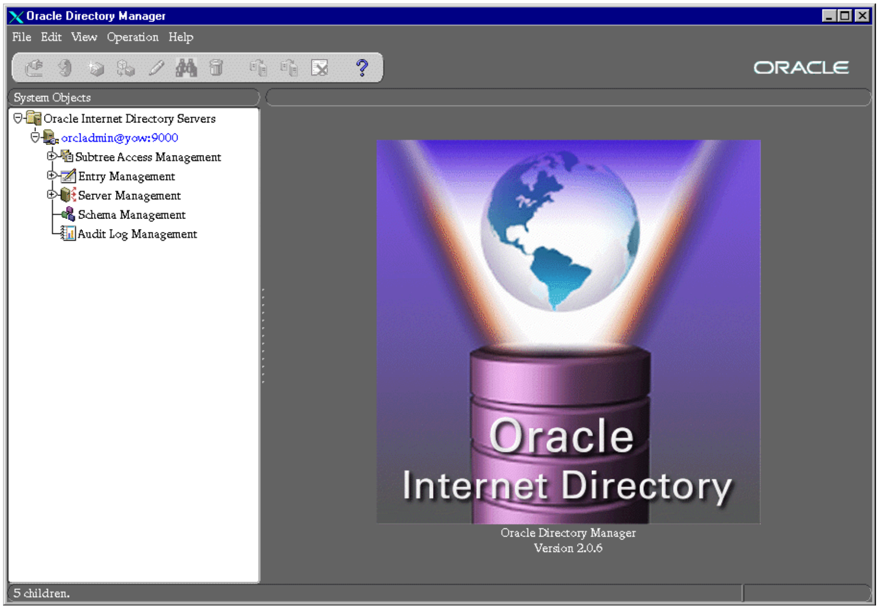
Table 4–2 SSL Information

Field	Description
SSL Location	The location of the user’s wallet. If the user’s wallet is on the local machine, enter the wallet’s path and file name. If the wallet is on another machine, link to that location, then enter the wallet’s linked path and file name.
SSL Password	The password to open the user’s wallet
SSL Authentication	Options are: ■ 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, only SSL encryption / decryption will be used. ■ SSL Client and Server Authentication—Two-way authentication, that is, both client and server send certificates to each other. ■ SSL Server Authentication—One-way authentication, that is, only the directory server authenticates itself to the client. The directory server sends its certificate to the client.

Note: If the server requires two-way authentication, each Oracle Directory Manager user must have a unique wallet. If one-way authentication is specified, several Oracle Directory Manager users can use a single wallet.

---

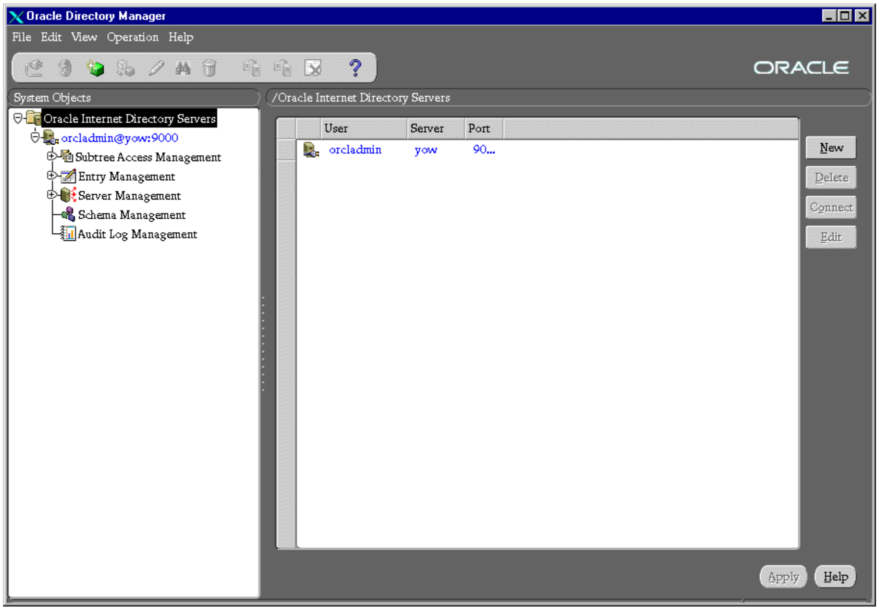
5. Click Login. Oracle Directory Manager appears:



Navigating 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. Tree items that have plus signs in front of them may have their own sub-tree items. The plus sign becomes a minus sign (-) when the entry is expanded. You can expand and contract the tree by clicking the plus signs and minus signs.

For example, if you click the plus sign next to Oracle Internet Directory Servers in the opening window navigator pane, the tree expands to show the connection information for the server to which you are connected:



User	Server	Port
orcladmin	yow	90...

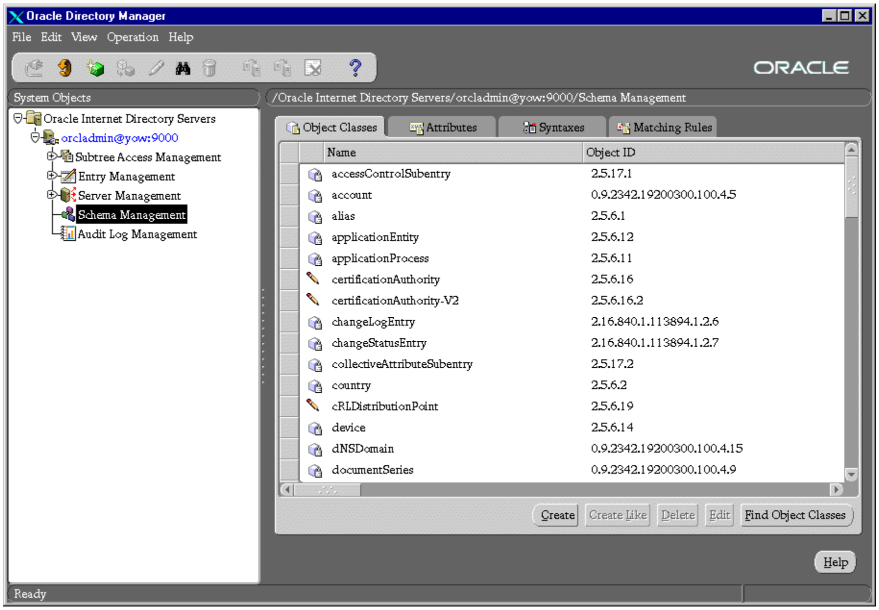
By clicking the plus sign (+) next to the server information, you make a number of other choices appear in the navigator pane.

You can navigate around Oracle Directory Manager using one or a combination of the following options.

- Select menu items from the menus across the top of the window.
- Use buttons on the toolbar across the top of the left pane to perform such actions as create, create-like, and delete.
- Click the tree items in the navigator pane in combination with the options in the right pane. For example, as shown in Figure 4–1, clicking the navigator tree item Schema Management causes its associated tab pages to display in the right pane. You can view and manipulate information on the Schema Management's four tab pages by selecting tabs in the right pane.
- Some right panes also include buttons you can click to start a task. For example, in the Schema Management pane (Figure 4–1), buttons are available to create a

new object, create a new object by copying an existing one (Create-Like), delete a selected object, edit a selected object, or find an object.

Figure 4–1 Schema Management Tab Pages



Name	Object ID
accessControlSubentry	2.5.17.1
account	0.9.2342.19200300.100.4.5
alias	2.5.6.1
applicationEntity	2.5.6.12
applicationProcess	2.5.6.11
certificationAuthority	2.5.6.16
certificationAuthority-V2	2.5.6.16.2
changeLogEntry	2.16.840.1.113894.1.2.6
changeStatusEntry	2.16.840.1.113894.1.2.7
collectiveAttributeSubentry	2.5.17.2
country	2.5.6.2
cRLDistributionPoint	2.5.6.19
device	2.5.6.14
dNSDomain	0.9.2342.19200300.100.4.15
documentSeries	0.9.2342.19200300.100.4.9

- Buttons at the bottom of the pane affect the whole pane. For example, in Figure 4–1, the Help button at the bottom of the pane displays the online help topic for the entire pane.

> **Note:** 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, the changes you have made in that window do not take effect, 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

Table 4–3 lists the menus you can access by using the menu bar, and describes the items in each menu. Menu items become enabled or disabled depending on the pane or tab page you are displaying in Oracle Directory Manager.

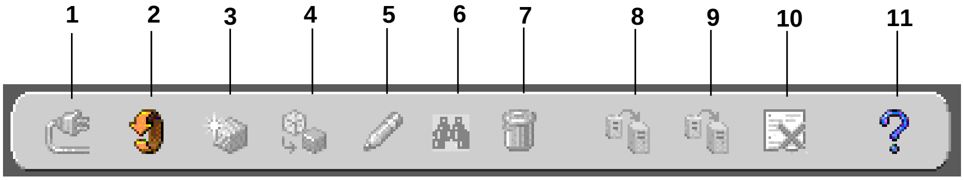
Table 4–3 Menu Bar

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
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. Useful when comparing two pieces of information.
Operations	Create Object Class—Displays the New Object Class dialog box which 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 which you use to add a new **Access Control Policy Point**. For information on ACPs, see Chapter 9. Create Entry—Displays the New Entry dialog box which you use to add a new directory entry Configure Entry Management—Displays a dialog box with fields to set the maximum number of subentries and the maximum search time for entry management.
Help	Contents—Displays the Contents tab page of the Help navigator Search for Help On...—Displays the Help Search dialog box which 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 4–2 and the accompanying table illustrate and describe the Oracle Internet Directory toolbar. Buttons become enabled or disabled depending on the pane or tab page you are displaying in Oracle Directory Manager.

Figure 4–2 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—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 Objects—Searches for an object
7	Remove—Removes an object
8	Refresh Entry—Updates data for entries stored in memory to reflect changes in the database
9	Refresh Subentries—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	Help—Displays the Help navigator

Connecting to Additional Directory Servers

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

To connect to an additional directory server:

1. In the navigator pane, expand 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 4-3.

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`. For example, on a Solaris platform, the path name would be `/$HOME/adsadmin.ini`.

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

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 6-5 "Managing Attributes by Using Oracle Directory Manager" on page 6-18
Entries management	"Managing Entries by Using Oracle Directory Manager" on page 7-2
ACP administration	"Managing Access Control by Using Oracle Directory Manager" on page 9-16
Partitioning and replication	Chapter 10

Using Command Line Tools

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

The command line tools act on entries that are in text files written in the LDAP Data Interchange Format (LDIF). An entry in the input file lists the DN, its attribute types (there may be many), and their values, with one attribute type per line.

> **See Also:** "Using LDAP Data Interchange Format (LDIF)" on page A-2 before actually formatting an input 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 for it.

Tool	Task(s)	Syntax and Usage Notes
ldapsearch	Search for directory entries	"ldapsearch" on page A-4
ldapbind	Authenticate user/client to a directory server	"ldapbind" on page A-6
ldapadd	Add entries one at a time	"ldapadd" on page A-7
ldapaddmt	Add several entries concurrently by using this multi-threaded tool	"ldapaddmt" on page A-9
ldapmodify	Create, update, and delete attribute data for an entry	"ldapmodify" on page A-11
ldapmodifymt	Modify several entries concurrently by using this multi-threaded tool	"ldapmodifymt" on page A-14
ldapdelete	Delete entries.	"ldapdelete" on page A-16
ldapcompare	See whether an entry contains a specified attribute value	"ldapcompare" on page A-17
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" on page A-18

See Also:

- "Using NLS with Command Line Tools" on page 11-5 for a discussion of command line tools and NLS
- "Using Bulk Tools" on page 4-14 for information on bulk command line tools

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 oidpassword tool.

See Also: "Using the OID Database Password Utility" on page 4-16

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

Tool	Task(s)	Syntax and Usage Notes
bulkload	Load large number of entries to Oracle Internet Directory through LDIF files	"bulkload" on page A-20
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" on page A-21
bulkmodify	Modify a large number of existing entries efficiently	"bulkmodify" on page A-23
bulkdelete	Delete a subtree efficiently	"bulkdelete" on page A-25

Using OID Control Utility

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

See Also:

- "Step Two: Start Server Instances" on page 3-3
- "Oracle Internet Directory Architecture" on page 2-27 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 by using the Catalog Management tool.

See Also: "Using the Catalog Management Tool" on page A-26 for syntax and usage notes

Using the OID Database Password Utility

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:** "Using the OID Database Password Utility" on page A-27 for syntax and usage notes

Administration Tasks at a Glance

Oracle Internet Directory administration tasks are described throughout this manual. Table 4–4 points you to the information you need for some of the more common tasks.

Table 4–4 Common Administration Tasks and Where To Find Instructions

Task	Information
Managing Attributes	
Add, modify, or delete an attribute by using command line tools	"Managing Attributes by Using Command Line Tools" on page 6-29
Add, modify, or delete an attribute by using the Oracle Directory Manager	"Managing Attributes by Using Oracle Directory Manager" on page 6-18
Managing Entries	
Add, modify, or delete a directory entry by using command line tools	"Managing Entries by Using Command Line Tools" on page 7-23
Add, modify, or delete a directory entry by using Oracle Directory Manager	"Managing Entries by Using Oracle Directory Manager" on page 7-2
Import bulk data files	"bulkload" on page A-20 "Using LDAP Data Interchange Format (LDIF)" on page A-2
View Directory Information Tree (DIT) hierarchy of entries	"Managing Entries by Using Oracle Directory Manager" on page 7-2
Managing Object Classes	
Add, modify, or delete object classes by using command line tools	"Managing Object Classes by Using Command Line Tools" on page 6-15
Add, modify, or delete object classes by using Oracle Directory Manager	"Managing Object Classes by Using Oracle Directory Manager" on page 6-5

Table 4–4 Common Administration Tasks and Where To Find Instructions

Task	Information
Managing Security	
Set up an Access Control Policy Point (ACP)	Chapter 9
Set up security	Chapter 8
Managing Servers	
Configure server instance parameters by using command line tools	"Managing Server Configuration Set Entries by Using Command Line Tools" on page 5-11
Configure server instance parameters by using the Oracle Directory Manager	"Managing Server Configuration Set Entries by Using Oracle Directory Manager" on page 5-4
Connect to a directory by using Oracle Directory Manager	"Connecting to a Directory Server" on page 4-3 "Connecting to Additional Directory Servers" on page 4-12
Run the directory server processes	Chapter 3
Stop the directory server processes	Chapter 3
View system operational attribute**s**	"Setting System Operational Attributes" on page 5-14
Managing Replication	
Set up replication	Chapter 10

Part II

Managing Oracle Internet Directory

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

- Chapter 5, "Managing an Oracle Directory Server"
- Chapter 6, "Managing Directory Schema"
- Chapter 7, "Managing Directory Entries"
- Chapter 8, "Managing Secure Sockets Layer (SSL)"
- Chapter 9, "Managing Directory Access Control"
- Chapter 10, "Managing Directory Replication"
- Chapter 11, "Managing National Language Support (NLS)"

5

Managing an Oracle Directory Server

This chapter explains how to manage the Oracle Internet Directory processes using Oracle Directory Manager and command line tools.

> **See Also:** Chapter 3 for instructions on starting and stopping Oracle directory server instances

The administration tasks are explained in the following sections:

- Managing Server Configuration Set Entries
- Setting System Operational Attributes
- Managing Super, Guest, and Proxy Users
- Viewing Active Server Instance Information
- Setting Debug Logging Levels by Using the OID Control Utility
- Using Audit Log
- Changing the Password to an Oracle Data Server

Managing Server Configuration Set Entries

When you issue a start-server message through the OID Control Utility, that 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 2-32 for a conceptual overview of configuration set entries
- "Step Two: Start Server Instances" on page 3-3 for instructions on how to issue the start-server message through the OID Control Utility

This section covers the following topics:

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

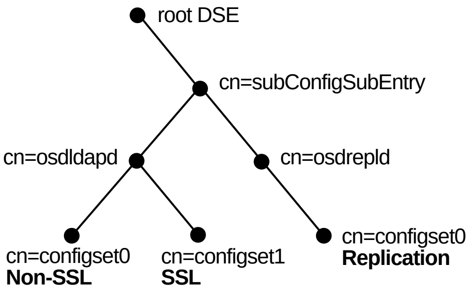
Preliminary Considerations

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 2.0.6, this applies to the LDAP 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 configset for groups with special needs.

Figure 5–1 Directory Entry Hierarchy Showing Multiple Configuration Set Entries



For example, Figure 5–1 shows:

- An LDAP server with one instance listening on the default port and using `configset0` with SSL set to *off*
- A second LDAP server instance listening on the SSL port and using `configset1` with `SSLenable` set to *on*
- A replication server instance using `configset0`

See Also:

- Chapter 8 for information about configuration parameters for SSL
- Chapter 10 for information about configuration parameters for replication
- Table F–1 on page F-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. These topics are covered in the following sections:

- Viewing Configuration Set Entries
- Adding Configuration Set Entries
- Modifying Configuration Set Entries
- Deleting 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 instances and restart them.

You can change a configuration set 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.

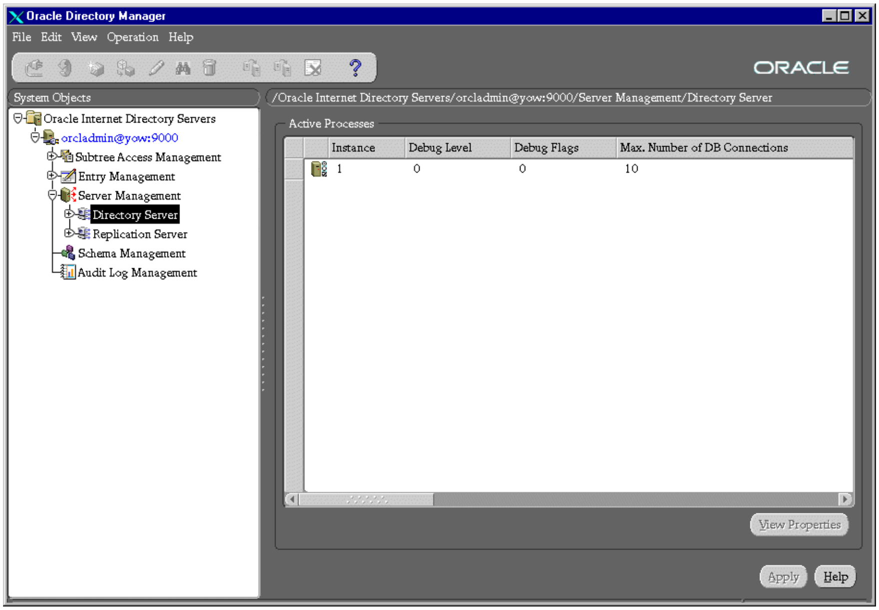
For information on restarting directory server instances, see "Restarting Directory Server Instances" on page 3-8.

Viewing Configuration Set Entries

To view configuration set entries:

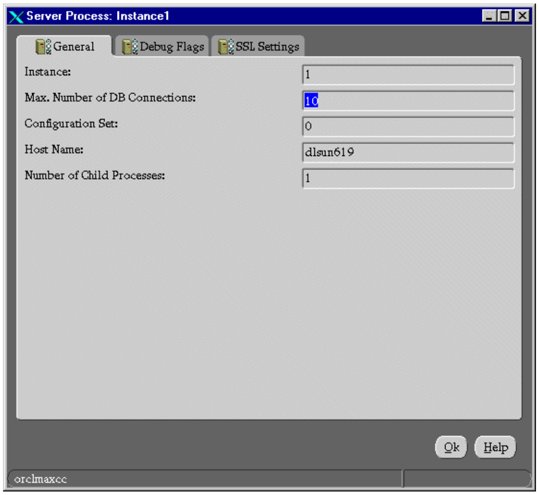
1. In the navigator pane, expand an instance, then expand Server Management.
2. Select Directory Server in the navigator pane.

The parameters of the active instance appear in the right pane. You can see all of them by scrolling horizontally.



Instance	Debug Level	Debug Flags	Max. Number of DB Connections
1	0	0	10

3. Click an instance in the right pane. A Server Process dialog box appears:



Instance:	1
Max. Number of DB Connections:	10
Configuration Set:	0
Host Name:	dlsun619
Number of Child Processes:	1

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.

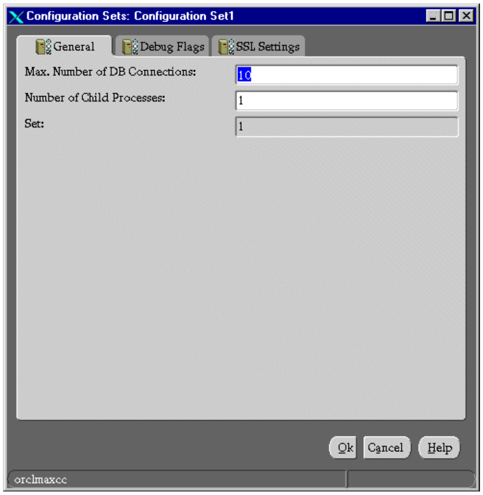
> **See Also:** "Modifying Configuration Set Entries" on page 5-10

Adding Configuration Set Entries

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.

To add configuration set entries:

1. In the navigator pane, expand Server Management > Directory Server or Replication Server. Select Default Configuration Set and, on the toolbar, click the Create Like button. The Configuration Sets dialog box displays the General tab:



Configuration Sets: Configuration Set1

Max. Number of DB Connections: 10

Number of Child Processes: 1

Set: 1

orclmaxcc

2. Fill in the fields with the information described in Table 5–1:

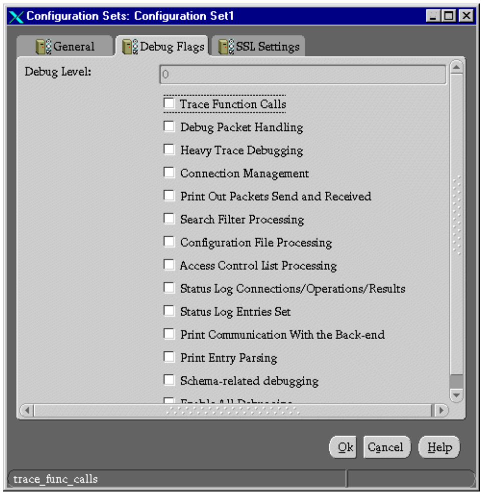
Table 5–1 Adding a Configuration Set Entry: General Tab

Field	Description
Max. Number of DB Connections	The number of concurrent database connections a single directory server process can have. The default is ten.

Table 5–1 Adding a Configuration Set Entry: General Tab

Field	Description
Number of Child Processes	The number of server processes a single instance can spawn. The default is one.
Set	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 Debug Flags tab:



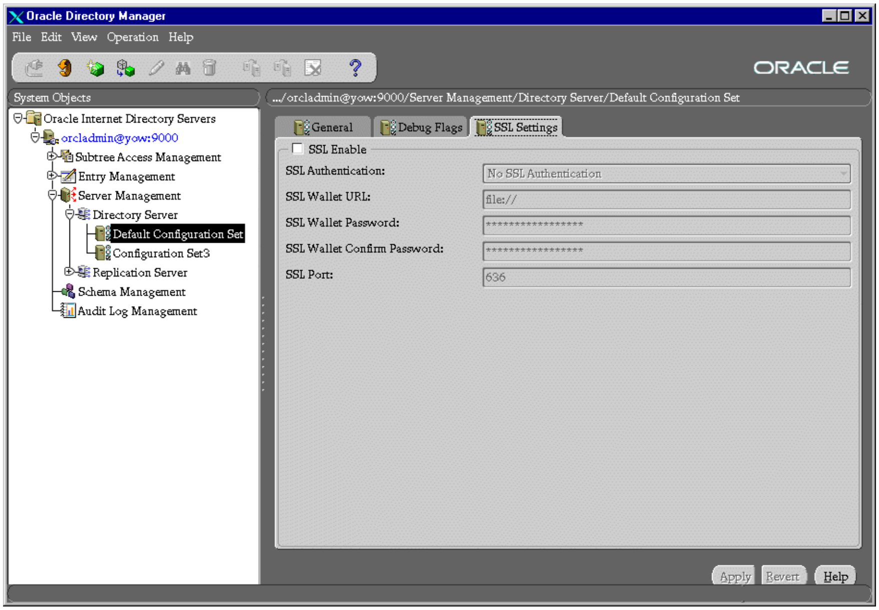
Debug Level: 0

☐ Trace Function Calls
☐ Debug Packet Handling
☐ Heavy Trace Debugging
☐ Connection Management
☐ Print Out Packets Send and Received
☐ Search Filter Processing
☐ Configuration File Processing
☐ Access Control List Processing
☐ Status Log Connections/Operations/Results
☐ Status Log Entries Set
☐ Print Communication With the Back-end
☐ Print Entry Parsing
☐ Schema-related debugging
☐ Enable All Debugging

trace_func_calls

Ordinarily, you can leave these radio buttons unselected. However, if you need to generate a log for a specific problem, you can use this tab page to specify the debug logging level.

4. Select the SSL Settings tab:



.../orcladmin@yow:9000/Server Management/Directory Server/Default Configuration Set

□ SSL Enable

SSL Authentication: No SSL Authentication

SSL Wallet URL: file://

SSL Wallet Password: *****************

SSL Wallet Confirm Password: *****************

SSL Port: 636

Fill in the fields with the information described in the following table:

Table 5–2 Adding a Configuration Set Entry: SSL Settings Tab

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, only SSL encryption / decryption is used. ■ SSL Client and Server Authentication—Both client and server authenticate themselves to each other. Both the client and server 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.

Table 5–2 Adding a Configuration Set Entry: SSL Settings Tab

Field	Description

SSL Wallet URL	Enter the location of the SSL 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/my_wallet`
	On Windows NT, you could set this parameter as follows:
	`file:C:\my_dir\my_wallet`
	For information on setting the location of the Oracle Wallet and the Oracle Wallet password, see Appendix D.

SSL Wallet Password	Enter the password for the wallet. This password was set during creation of the wallet. See "Creating a New Wallet" on page D-6. 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 OK.

Note: Remember, the changes will not affect the active instance until you restart it. See "Restarting Directory Server Instances" on page 3-8.

See Also: "Setting Debug Logging Levels by Using the OID Control Utility" on page 5-21

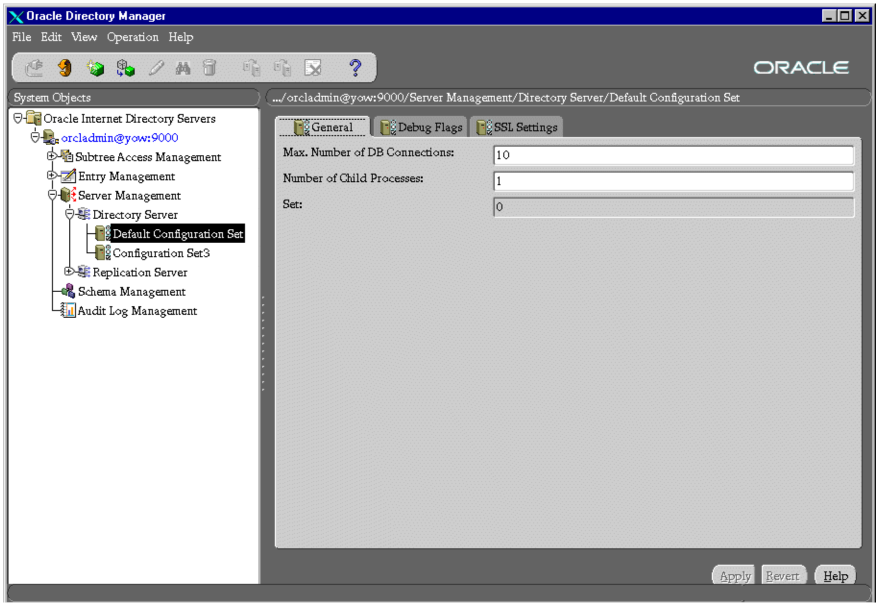
To create a new configuration set entry without copying from a previous entry:

1. In the navigator pane expand the server instance to which you are connecting, then expand Server Management > Directory Server > Default Configuration Set.
2. Select Default Configuration Set.
3. On the toolbar, click the Create button. A Configuration Sets dialog box displays the number of the new configuration set. Fill in the fields as described in Table 5–2 on page 5-8.

Modifying Configuration Set Entries

To modify configuration set entries:

1. In the navigator pane, expand Directory Server and select the configuration set entry you want to modify. The configuration set appears in the group of tab pages in the right pane:



.../orcladmin@yow:9000/Server Management/Directory Server/Default Configuration Set

Max. Number of DB Connections: 10

Number of Child Processes: 1

Set: 0

2. Modify the values in the fields for the General tab. For information about each field, see Table 5–1 on page 5-6. You can change any of the values. Press Apply to save the changes.
3. Select the Debug Flags tab. Select the check boxes for the debug logging levels you want to use.
4. Select the SSL Settings tab. Fill in the fields as required. For information about each field, see Table 5–2 on page 5-8.

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

Note: Remember, the changes will not affect the active instance until you restart it. See "Restarting Directory Server Instances" on page 3-8.

Deleting Configuration Set Entries

To delete configuration set entries:

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

Note: Remember, the changes will not affect the active instance until you restart it. See "Restarting Directory Server Instances" on page 3-8.

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 LDAP servers.

This section tells you how to perform the tasks described in the following sections:

- Adding Configuration Set Entries by Using ldapadd
- Modifying and Deleting Configuration Set Entries by Using ldapmodify

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: "Using LDAP Data Interchange Format (LDIF)" 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. These operations are explained in the steps below.

1. Create the input file in a text editor.

   Input files must use LDIF format, which is explained in "Using LDAP Data Interchange Format (LDIF)" on page A-2. 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 the following 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 the example above were given the file name `newconfigs`, the ldapadd command would look something like the following:

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

See Also:

- "ldapadd" on page A-7 for a detailed list of options available with this command
- Table F–1 on page F-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. These operations are explained in the steps below.

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. LDIF format is explained in "Using LDAP Data Interchange Format (LDIF)" on page A-2.

   In the example of the input file 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.

   See Also: Table F–1 on page F-5 for a description of configuration set entry attributes

   ```
   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 below:

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

   See Also: "ldapmodify" on page A-11 for a more detailed discussion of ldapmodify, and a list of its options

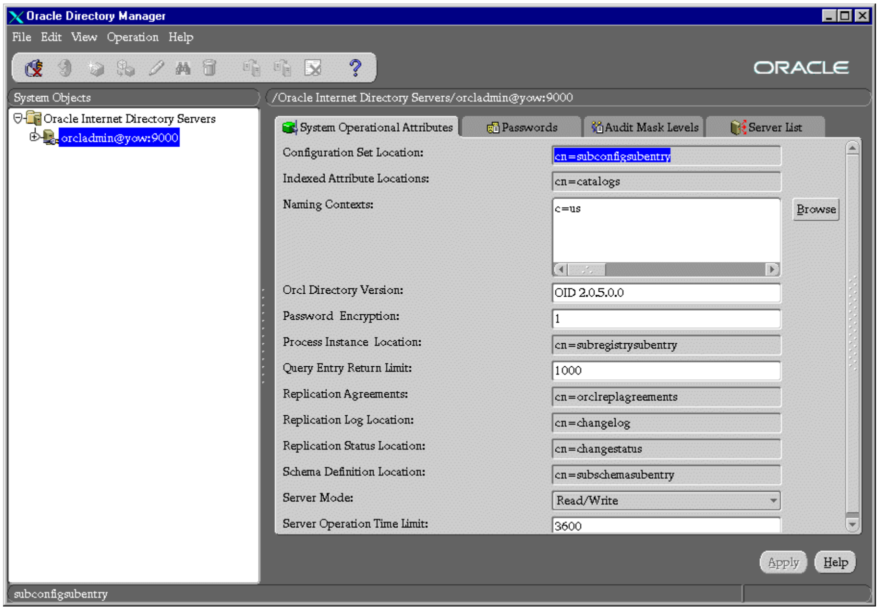
Setting System Operational Attributes

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.

See Also: "Kinds of Attribute Information" on page 2-4.

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:



Oracle Directory Manager
File Edit View Operation Help
ORACLE
System Objects
Oracle Internet Directory Servers
orcladmin@yow:9000
/Oracle Internet Directory Servers/orcladmin@yow:9000
System Operational Attributes
Passwords
Audit Mask Levels
Server List
Browse
Apply
Help
subconfigsubentry

Configuration Set Location: cn=subconfigsubentry
Indexed Attribute Locations: cn=catalogs
Naming Contexts: c=us
Orcl Directory Version: OID 2.0.5.0.0
Password Encryption: 1
Process Instance Location: cn=subregistrysubentry
Query Entry Return Limit: 1000
Replication Agreements: cn=orclreplagreements
Replication Log Location: cn=changelog
Replication Status Location: cn=changestatus
Schema Definition Location: cn=subschemasubentry
Server Mode: Read/Write
Server Operation Time Limit: 3600

Table 5–3 describes the field for each operational attribute.

Table 5–3 Operational Attribute Fields

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 context(s) contained in this server. Enter a new value in the field. If you are not sure of the value, click the attribute to bring up a search window.	none	Yes
Password Encryption	Determines whether the password is stored in encrypted form. 0=No. 1=Yes.	1	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
Server Mode	Determines whether data can be written to the server. 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

Setting System Operational Attributes by Using ldapmodify

The modifiable system operational attributes are:

Attribute	Description	Default
`namingContexts`	DN for the naming context(s) contained in this server. Enter a new value in the field. If you are not sure of the value, click the attribute to bring up a search window.	none
`orclUseEncrypt`	Determines whether the password is stored in encrypted form. 0=No. 1=Yes.	1
orclSizeLimit	Maximum number of entries to be returned by a search	1000
`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" on page A-11 for a more detailed discussion of ldapmodify, and a list of its options

Managing Super, Guest, and Proxy Users

A super user is a special directory administrator who typically has completely open privileges to directory information.

A guest user is one who is not an anonymous user, and, at the same time, does not have a specific user entry.

A proxy user is typically used in an environment with a middle tier such as a firewall. In such an environment, the end user authenticates himself 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.

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 onto 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 9 for information on how to set access rights

This section covers topics in the following subsections:

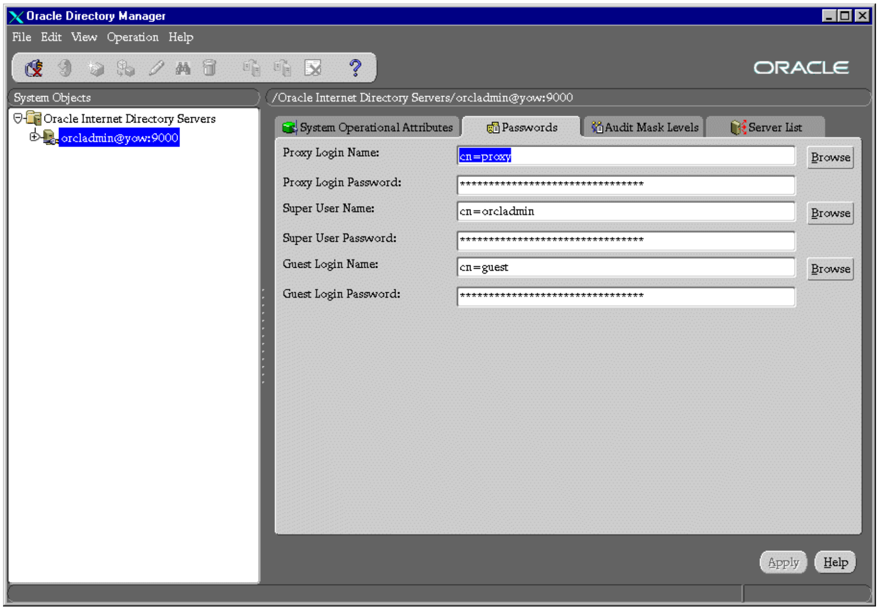
- "Managing User Names and Passwords by Using Oracle Directory Manager"
- "Managing User Names and Passwords by Using ldapmodify"

Managing User Names and Passwords by Using Oracle Directory Manager

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

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

1. In the navigator pane, expand Oracle Internet Directory Servers.
2. Select the server. The group of tab pages for that server appear in the right pane.
3. Select the Passwords tab. The Password tab page displays the current user names and passwords for each type of user. Note that passwords are not displayed in the password fields.


Oracle Directory Manager
File Edit View Operation Help
ORACLE
System Objects
Oracle Internet Directory Servers
orcladmin@yow:9000
/Oracle Internet Directory Servers/orcladmin@yow:9000
System Operational Attributes
Passwords
Audit Mask Levels
Server List
Browse
Browse
Browse
Apply
Help

Proxy Login Name: cn=proxy

Proxy Login Password: **********************************

Super User Name: cn=orcladmin

Super User Password: **********************************

Guest Login Name: cn=guest

Guest Login Password: **********************************

Table 5–1 lists and describes the fields in the Passwords tab page.

Table 5–4 Password Fields and Descriptions

Field	Description
Super User Name	The default is `cn=orcladmin`.
Super User Password	The default is `welcome`. You should change this password immediately.
Guest Login Name	Guests have privileges determined by the Access Control Policy Points (ACPs) in the directory. The default is `cn=guest`.
Guest Login Password	The default is `guest`.
Proxy Login Name	Proxy users have privileges determined by the ACPs in the directory. The default is `cn=proxy`.
Proxy Login Password	The default is `proxy`.

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

Managing User Names and Passwords by Using ldapmodify

To change administrative user names and passwords, you use ldapmodify on the following attributes.

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

See Also: "ldapmodify" on page A-11 for ldapmodify syntax and usage notes.

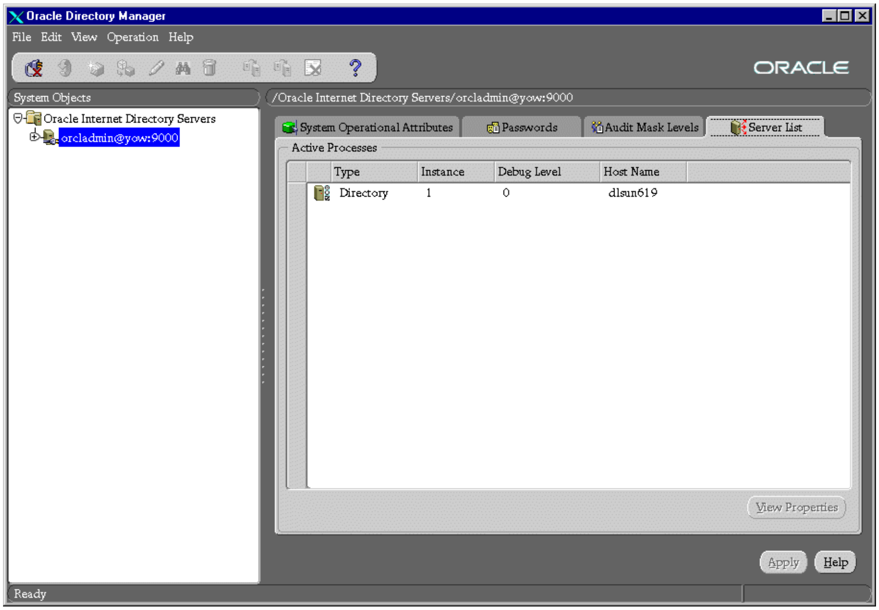
For example, to change the password of the super user to `superuserpassword`, we would use ldapmodify to modify the DSE by using an LDIF file containing the following:

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

Viewing Active Server Instance Information

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

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 Server List tab to display basic information—namely, type, instance number, debug level, and host name—for all active server instances:


Oracle Directory Manager
File Edit View Operation Help
ORACLE
System Objects
Oracle Internet Directory Servers
orcladmin@yow:9000
/Oracle Internet Directory Servers/orcladmin@yow:9000
System Operational Attributes
Passwords
Audit Mask Levels
Server List
Active Processes
View Properties
Apply
Help
Ready

Type	Instance	Debug Level	Host Name
Directory	1	0	dlsun619

4. To see configuration parameters for a particular server instance, in the Server List tab page, select the server.
5. Click View Properties. The Server Process dialog box displays configuration parameters for the 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 5-4 for instructions on changing configuration set entries

Setting Debug Logging Levels by Using the OID Control Utility

This section tells you how to set debug logging levels by using the OID Control Utility. You can also set debug logging levels by using Oracle Directory Manager as described in "Managing Server Configuration Set Entries by Using Oracle Directory Manager" on page 5-4.

To set debug logging levels by using the OID Control Utility, restart the Oracle Directory Server using the -debug option for an LDAP server, and the -d flag for the replication server. Use the debug level number based on Table 5–5.

Since 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.

For example, if you want to trace function calls (1) and active connection management (8), you would 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
```

The above example restarts the LDAP server as well as the replication server with the debugging flags. Table 5–5 provides the complete list of debug logging levels.

Table 5–5 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

Table 5–5 Debug Logging Levels

Logging Level Value	Function
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

Using Audit Log

The audit log records critical events on the Oracle Directory Server that are important from a security point of view or for operations. An administrator can query the audit log using ldapsearch commands. Because the log generation is contingent upon events occurring on the server, only the Oracle Internet Directory server itself can create the log entries. You cannot add audit log entries with either the Oracle Directory Manager or the command line tools. Only the server can add entries.

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.

See Also:

- "Searching for Audit Log Entries" on page 7-8
- "Searching for Audit Log Entries by Using ldapsearch" on page 5-28

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

To clean up audit log entries, use bulkdelete to remove all the audit log entries. Specify `cn=auditlog` as the base of the bulkdelete. Because bulkdelete will delete all the entries under cn=auditlog, use LDIF writer to write the entries to a file for later reference.

> **See Also:** "bulkdelete" on page A-25

The remainder of this section discusses topics in the following subsections:

- Structure of Audit Log Entries
- Position of Audit Log Entries in the DIT
- Auditable Events
- Auditing Events
- Setting the Audit Level by Using Oracle Directory Manager
- Setting the Audit Level by Using ldapmodify
- Searching for Audit Log Entries

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 the following:

Attribute	Description
orclsequence	Used to create the name of the entry. The name is generated using a database sequence.
orcleventtype	Type of event that occurred. This is a catalogued attribute.
orcleventtime	The time at which the event occurred.
orcluserdn	Identity of the user who logged into the Oracle Internet Directory server to perform the operation. This attribute is catalogued.
orclopresult	Outcome of the operation. SUCCESS if the operation succeeded, or else the reason why the operation failed.
orclauditmessage	The textual message. This attribute is not catalogued.
objectclass	Has the preset values top and orclauditoc.

See Also: "Object Class Types" on page 2-9 for a description of `top`

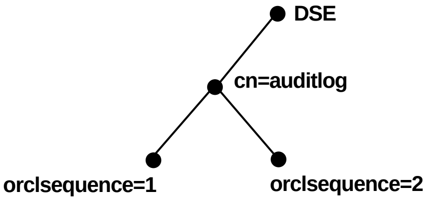
Note that the audit log entries do not become part of a regular search result set even though the search filter may satisfy the query criteria. For example, a search with the condition `objectclass=top` will not yield results from the auditlog entries. Only a search with `cn=auditlog` as the base of the search will 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 6-31.

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 5–2.

Figure 5–2 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

Table 5–6 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. This is explained in "Setting the Audit Level by Using ldapmodify" on page 5-27.

Table 5–6 Auditable Events

Event	Description	Audit Levels
Super user login	Super user bind to the server (success and failures)	0x0001
Schema element add/replace	Adding a new schema element (success and failure)	0x0002
Schema element delete	Deleting a schema (success and failures)	0x0004
Bind	Unsuccessful bind cases	0x0008
Access violation	Access denied by **ACP**	0x0010
DSE modification	Changes to DSE entry (success and failures)	0x0020
Replication login	Replication server authentication (success and failures)	0x0040
ACL modification	Changes to ACPs	0x0080
User password modification	Modification of user password attribute	0x0100
Add	ldapadd operation (success and failures)	0x0200
Delete	ldapdelete operation (success and failures)	0x0400
Modify	ldapmodify operation (success and failures)	0x0800
ModifyDN	ldapModifyDN operation (success and failures)	0x1000

Auditing Events

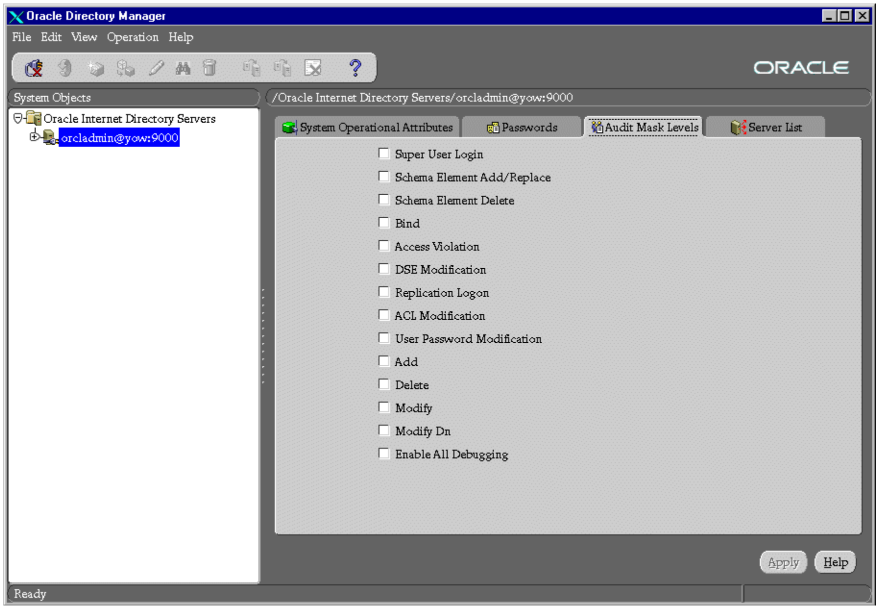
Events described in the previous section can be turned on or off. DSE attribute `orclauditlevel` indicates the current audit level set on the server. A value of 0 for the attribute means no auditing, which is the default.

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.
2. Select the server.
3. In the work pane, select the Audit Mask Levels tab page:


Oracle Directory Manager
File Edit View Operation Help
ORACLE
System Objects
Oracle Internet Directory Servers
orcladmin@yow:9000
/Oracle Internet Directory Servers/orcladmin@yow:9000
System Operational Attributes
Passwords
Audit Mask Levels
Server List
Apply
Help
Ready

☐ Super User Login
☐ Schema Element Add/Replace
☐ Schema Element Delete
☐ Bind
☐ Access Violation
☐ DSE Modification
☐ Replication Logon
☐ ACL Modification
☐ User Password Modification
☐ Add
☐ Delete
☐ Modify
☐ Modify Dn
☐ Enable All Debugging

4. Select the check box for the audit level you want to use.
5. Click Apply.

> **See Also:** Table 5–6 on page 5-25 for a description of each audit mask 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:

Schema element delete	0x0004	4
DSE modification	0x0020	32
Add	0x0200	512
		548

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

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

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

> **See Also:** "Restarting Directory Server Instances" on page 3-8

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 on page 7-8

Searching for Audit Log Entries by Using ldapsearch

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

See: "ldapsearch" on page A-4

Changing the Password to an Oracle Data 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: "Using the OID Database Password Utility" on page A-27

6

Managing Directory Schema

This chapter covers topics in the following sections:

Guidelines for Managing Object Classes

This section explains how to add and modify **object classes**. 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:

- Chapter 2 for a conceptual overview of LDAP schema components
- Appendix F for a list of schema components installed with Oracle Internet Directory

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 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: "Rules for Managing Attributes" on page 6-16.

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.

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 2-8.

Modifying Object Classes

Listed below are the types of modifications you can make to an existing object class. The rules for these modifications are explained later in this section. You can perform any modifications through Oracle Directory Manager and through the command line tools.

It is generally not a good idea to modify object classes, except auxiliary object classes. If existing object classes do not have the attributes you need, it is better to create an auxiliary object class and associate the needed attributes with it.

You can make the following 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 the following guidelines in mind:

- You cannot modify an object class that is part of the standard LDAP schema.
- 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 6-5
- "Managing Object Classes by Using Command Line Tools" on page 6-15.

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 discusses using Oracle Directory Manager to perform the administrative tasks described in the following sections:

- Searching for Object Classes
- Viewing Properties of Object Classes
- Adding Object Classes
- Modifying Object Classes
- Deleting Object Classes

Searching for Object Classes

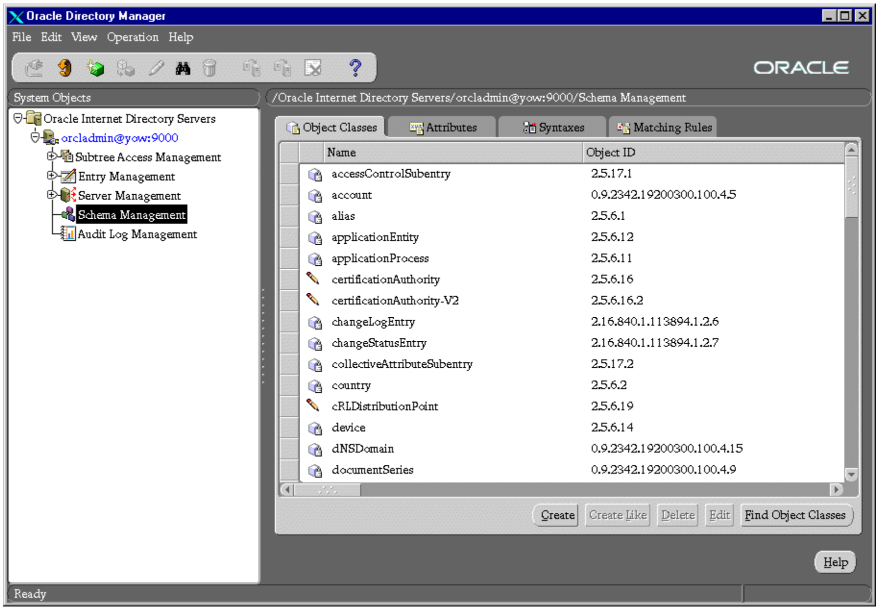
You can specify your search for an object class by:

- 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:

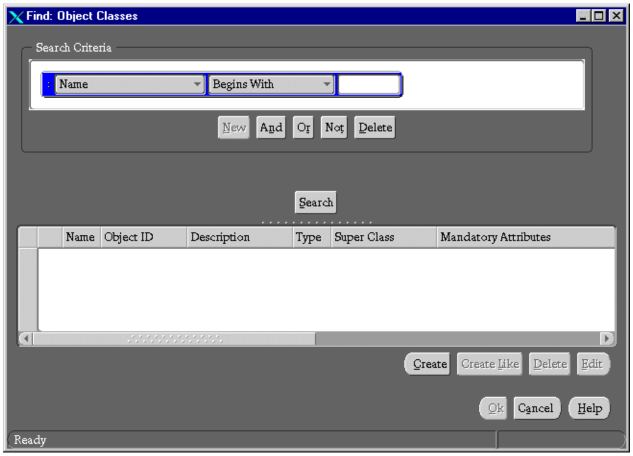
1. In the navigator pane, select Schema Management. The Schema Management tab pages appear in the right pane:



/Oracle Internet Directory Servers/orcladmin@yow:9000/Schema Management

Name	Object ID
accessControlSubentry	2.5.17.1
account	0.9.2342.19200300.100.4.5
alias	2.5.6.1
applicationEntity	2.5.6.12
applicationProcess	2.5.6.11
certificationAuthority	2.5.6.16
certificationAuthority-V2	2.5.6.16.2
changeLogEntry	2.16.840.1.113894.1.2.6
changeStatusEntry	2.16.840.1.113894.1.2.7
collectiveAttributeSubentry	2.5.17.2
country	2.5.6.2
cRLDistributionPoint	2.5.6.19
device	2.5.6.14
dNSDomain	0.9.2342.19200300.100.4.15
documentSeries	0.9.2342.19200300.100.4.9

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	The type of object class for which you are searching, whether abstract, structural, or auxiliary
Superclass	The 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 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 property begins with the letters `orcl`, type those letters in the text box at the right end of the search criteria bar.

5. 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	To search 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	To search 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.
Contains	To search 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	To search 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	To search 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	To search 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.

Filter	Description
Not Null	To search 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.

6. Below the Search Criteria field are five buttons described in Table 6–1. Use these buttons to further refine your search.

Table 6–1 Search Criteria Buttons

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

You can view properties of object classes as described in the following sections:

- Viewing All Object Classes in the Schema
- Viewing Properties of an Individual Object Class

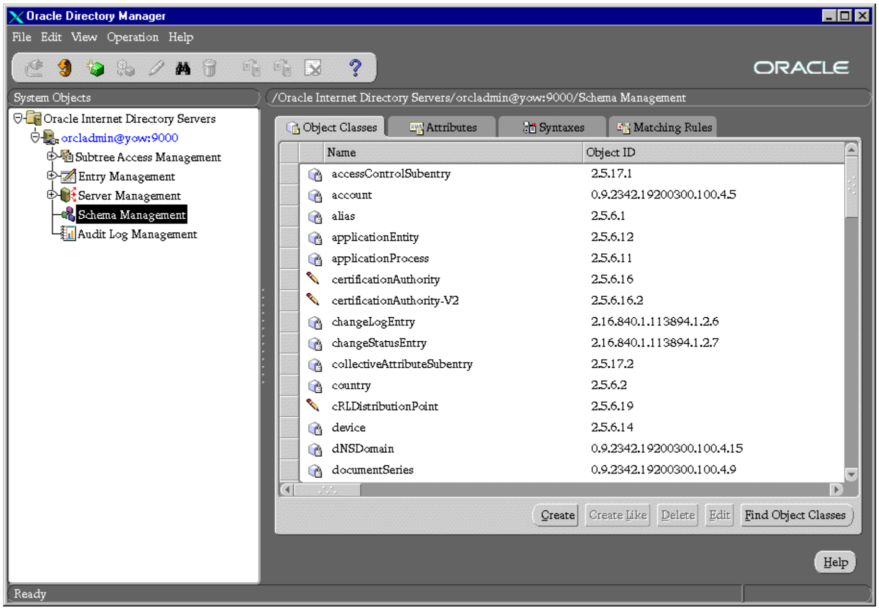
Viewing All Object Classes in the Schema

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. Click the Object Classes tab in the right pane. A pane like the following appears.

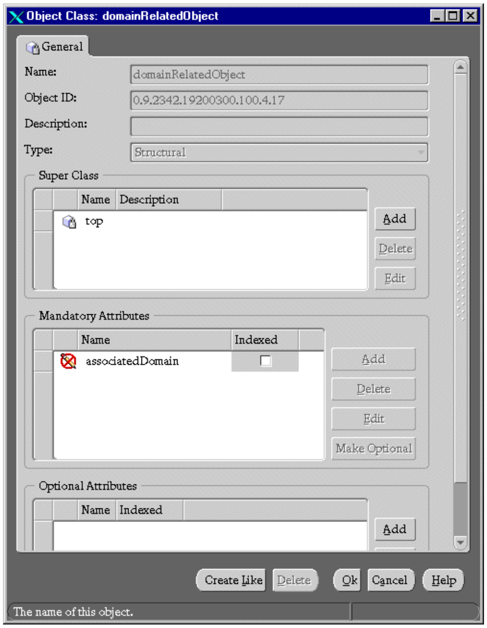


Name	Object ID
accessControlSubentry	2.5.17.1
account	0.9.2342.19200300.100.4.5
alias	2.5.6.1
applicationEntity	2.5.6.12
applicationProcess	2.5.6.11
certificationAuthority	2.5.6.16
certificationAuthority-V2	2.5.6.16.2
changeLogEntry	2.16.840.1.113894.1.2.6
changeStatusEntry	2.16.840.1.113894.1.2.7
collectiveAttributeSubentry	2.5.17.2
country	2.5.6.2
cRLDistributionPoint	2.5.6.19
device	2.5.6.14
dNSDomain	0.9.2342.19200300.100.4.15
documentSeries	0.9.2342.19200300.100.4.9

To see all the information in this pane, you can scroll horizontally and vertically.

Viewing Properties of an Individual Object Class

To examine an individual object class and its attributes, select the Object Classes tab and double-click the object class in the list displayed in the panel. The properties of the selected object class appear in the Object Class dialog box:



Object Class: domainRelatedObject

General

Name: domainRelatedObject

Object ID: 0.9.2342.19200300.100.4.17

Description:

Type: Structural

Super Class

Name	Description
top	

Mandatory Attributes

Name	Indexed
associatedDomain	□

Optional Attributes

Name	Indexed

The name of this object.

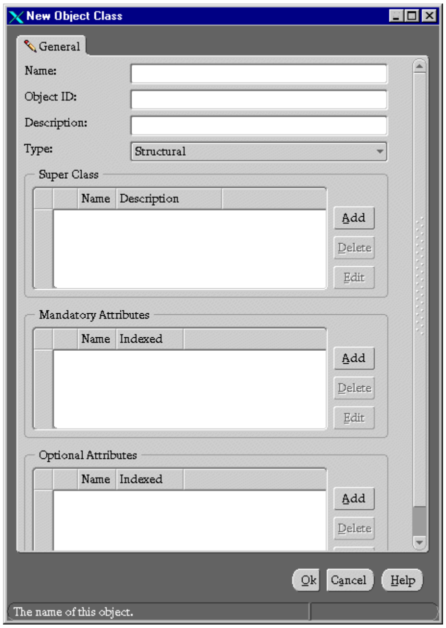
Object classes from which attributes may be inherited are listed in the Super Class pane. The mandatory attributes are listed in the Mandatory Attributes window and optional attributes are listed in the Optional Attributes window. Each window indicates whether the attributes are indexed so that they can be used in a search expression.

Adding Object Classes

To add object classes by using Oracle Directory Manager:

1. In the navigator pane, select Schema Management, then 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:



The name of this object.

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

2. Enter the information in the fields that are described in Table 6–2.

Table 6–2 Fields for Adding a New Object Class

Field	Description
Name	Name of the object class you are creating
Object ID	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. If you are creating a new object class, assign an identifier that is sure to be unique, following the system agreed upon within your organization.
Description	This optional field is for your information only
Type	The type of object class: Abstract, Structural, Auxiliary, None. **See Also:** "Object Class Types" on page 2-9
Super Class	The class(es) from which you are deriving this new object class. The new object class will inherit all the attributes of the superclass(es) you select. All structural object classes must have `top` as one of its superclasses. **See Also:** "Subclasses, Superclasses, and Inheritance" on page 2-8
Mandatory Attributes	Attributes for which values *must* be entered
Optional Attributes	Attributes for which values *may* be entered

You can add objects by clicking the buttons to the right of each window.

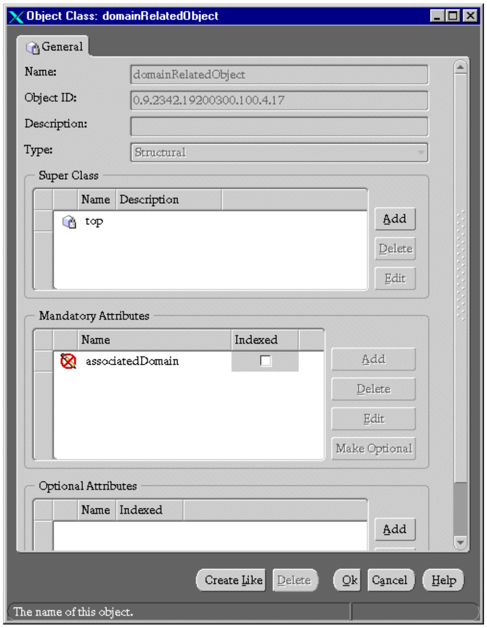
3. Click OK.

> **See Also:** The online help for further details on adding object classes by using Oracle Directory Manager

Modifying Object Classes

To modify an existing object class:

1. In the navigator pane, expand Schema Management.
2. Select the Object Classes tab page.
3. In the Object Classes tab page, double-click the object class you want to modify. The Object Class dialog box appears:



Object Class: domainRelatedObject

General

Name: domainRelatedObject

Object ID: 0.9.2342.19200300.100.4.17

Description:

Type: Structural

Super Class

Name	Description
top	

Mandatory Attributes

Name	Indexed
associatedDomain	☐

Optional Attributes

Name	Indexed

The name of this object.

4. Enter the information in the fields described in Table 6–2 on page 6-13.

   You can add attributes by clicking the buttons to the right of each window.

5. Click Apply.

> **See Also:** Oracle Directory Manager online help for further details on modifying object classes

Deleting Object Classes

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.

To delete an object class:

1. In the navigator pane, select Schema Management.
2. Select the Object Classes tab.
3. In the Object Classes tab page, select the object class you want to delete.
4. 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" on page A-11

Example 1: Adding a New Object Class

To add a new object class schema component by using ldapmodify, at the system prompt type a command using the following syntax:

```
ldapmodify -h host -p port -f ldif_filename
```

In this example, the LDIF input file 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 ) )
```

The example above 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.

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

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

Example 2: Modifying an Auxiliary Object Class by Adding a New Attribute

To modify an auxiliary object class by adding a new attribute, use ldapmodify. 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 )  )
```

Rules for Managing Attributes

This section explains how to add, modify, and delete user-defined 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.

See Also: "Attributes" on page 2-3

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.

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.

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

Oracle Directory Manager allows you to manage attributes by performing tasks described in the following sections:

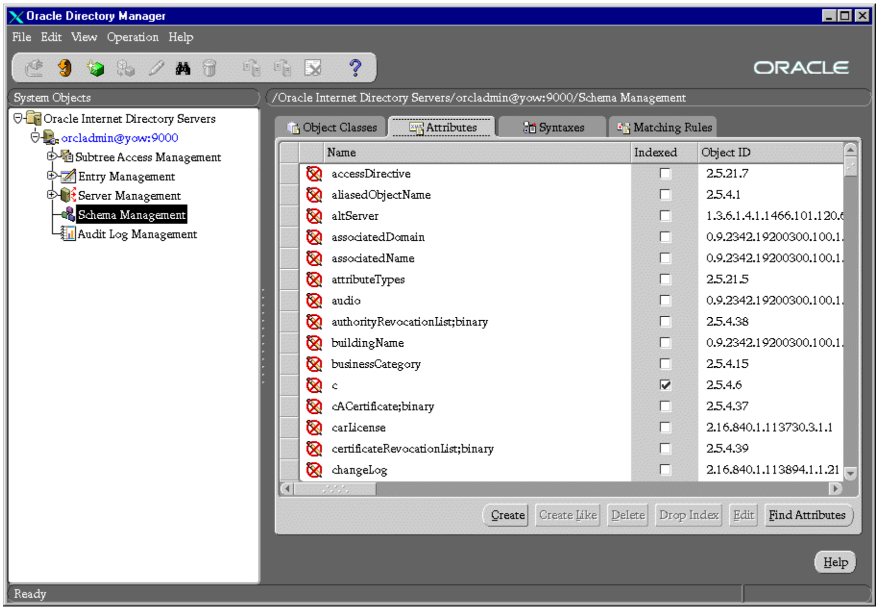
- Searching for Attributes
- Adding an Attribute
- Modifying an Attribute
- Indexing an Attribute

Searching for Attributes

To search for attributes by using Oracle Directory Manager:

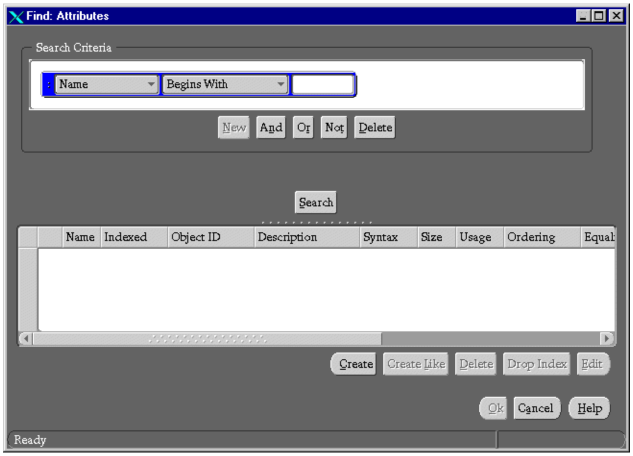
1. In the navigator pane, select Schema Management. The Schema Management tab pages appear in the right pane.

2. Select the Attributes tab page:



Name	Indexed	Object ID
accessDirective	□	2.5.21.7
aliasedObjectName	□	2.5.4.1
altServer	□	1.3.6.1.4.1.1466.101.120.6
associatedDomain	□	0.9.2342.19200300.100.1.
associatedName	□	0.9.2342.19200300.100.1.
attributeTypes	□	2.5.21.5
audio	□	0.9.2342.19200300.100.1.
authorityRevocationList;binary	□	2.5.4.38
buildingName	□	0.9.2342.19200300.100.1.
businessCategory	□	2.5.4.15
c	☑	2.5.4.6
cACertificate;binary	□	2.5.4.37
carLicense	□	2.16.840.1.113730.3.1.1
certificateRevocationList;binary	□	2.5.4.39
changeLog	□	2.16.840.1.113894.1.1.21

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	Indicates that this attribute type contains a maximum of one value
Super	Super attribute for the attribute for which you are searching

5. 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`.

6. 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	To search 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	To search 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	To search 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	To search 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	To search 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	To search 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	To search 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.

7. Beneath the Search Criteria field are five buttons described in the table below. 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

You can use Oracle Directory Manager to add attributes as described in the following sections:

- Adding a New Attribute
- Adding an Attribute by Copying an Existing Attribute

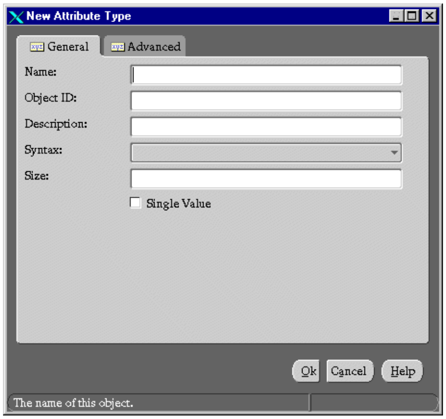
> **Tip:** Since equality, syntax, and matching rules are numerous and complex, it may be simpler to copy these characteristics from a similar existing attribute.

Adding a New Attribute

To add a new attribute by using Oracle Directory Manager:

1. In the navigator pane, select the Schema Management tab.
2. Choose one of the following methods:
   - In the right pane, select the Attributes tab, then click the Create button in the toolbar.
   - Click the Create button at the bottom of the right pane.
   - Select Create Attribute from the Operation menu.

The New Attribute Type dialog box appears:



The name of this object.

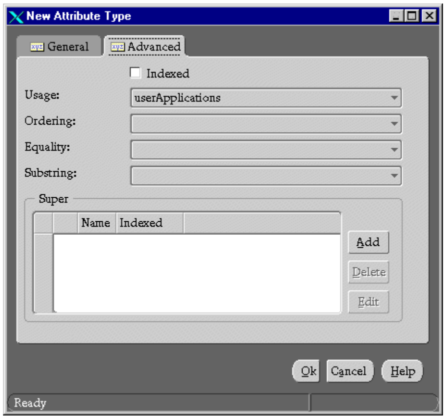
This dialog box contains two tab pages—General and Advanced—with fields in which you enter values either by typing or selecting from menus.

3. In the General tab, enter values in each of the fields as described in Table 6–3.

Table 6–3 Fields for Adding Attributes in General Tab

Field	Description
Name	Name for this attribute
Object ID	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 website.
Description	This optional field is for your information only.
Syntax	The standardized rules for data entry applicable to this attribute type.
Size	Maximum size allowed for this object
Single Value	Selecting this check box indicates that this attribute type contains a maximum of one value.

4. Select the Advanced tab:



New Attribute Type

□ Indexed

Usage: userApplications

Ordering:

Equality:

Substring:

In the Advanced tab page, enter values in each of the fields as described in Table 6–4.

Table 6–4 Fields for Adding Attributes in Advanced Tab

Field	Description
Indexed	Selecting the Indexed check box adds 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	Standards specifying 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	Standards specifying how precedence is established for values
Equality	Standards specifying how equality is determined in compare and search operations

Table 6–4 Fields for Adding Attributes in Advanced Tab

Field	Description
Substring	Used for regular expression matching
Super	Super attribute for this attribute. To add the super attribute, click the Add button next to this field. The Super Attribute Selector appears. Select the super attribute and click Select. Repeat as needed. To delete a super attribute from the Super field, select it, then click Delete.

5. Click OK.

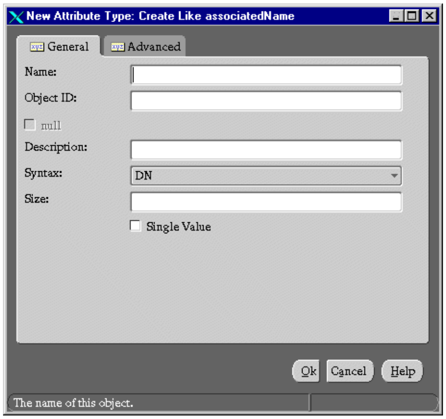
> **Note:** To use this attribute, remember to declare it to be part of the attribute set for an object class. You do this by selecting Schema Management in the navigator pane, then, in the right pane, selecting the Object Classes tab page. For further instructions, see "Modifying Object Classes" on page 6-3.

Adding an Attribute by Copying an Existing Attribute

To add an attribute by copying an existing attribute:

1. In the navigator pane, select Schema Management. The right pane displays the tab pages you use to manage the schema.
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 Attribute dialog box for that attribute appears:



New Attribute Type: Create Like associatedName

Name:

Object ID:

□ null

Description:

Syntax: DN

Size:

□ Single Value

The name of this object.

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 Table 6–3 on page 6-23. You must always change the DN to that of the new attribute.

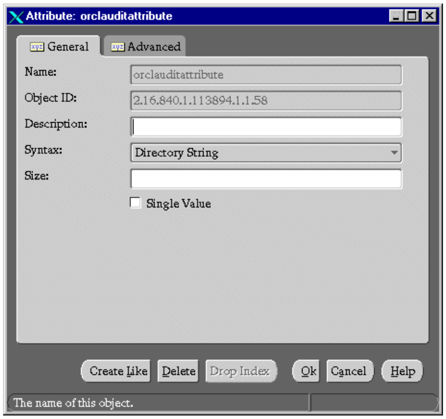
6. Select the Advanced tab and enter values in each of the fields as described in Table 6–4 on page 6-24.

7. Click OK to save your changes.

Modifying an Attribute

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 and double-click an editable attribute in the list. The Attribute dialog box displays properties of the selected attribute:



Attribute: orclauditattribute

Name: orclauditattribute

Object ID: 2.16.840.1.113894.1.1.58

Description:

Syntax: Directory String

Size:

□ Single Value

The name of this object.

This dialog box contains two tab pages—General and Advanced—with fields in which you enter values either by typing or selecting from menus.

3. Select the General tab and enter values in each of the fields as described in Table 6–3 on page 6-23.
4. Select the Advanced tab and enter values in each of the fields as described in Table 6–4 on page 6-24.
5. Click OK.

Indexing an Attribute

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 cannot use Oracle Directory Manager to index an already existing attribute. If you are using Oracle Directory Manager, you can index an attribute only at the time when you create it. To index an already existing attribute, use the Catalog Management tool.

Also, only those attributes that have an equality matching rule can be indexed.

See Also: "Indexing an Attribute by Using Command Line Tools" on page 6-31 for instructions on using the command line catalog management tool

This section covers topics in the following subsections:

- Viewing Indexed Attributes
- Indexing an Attribute When You Create It
- Dropping an Index from an Attribute

Viewing Indexed Attributes

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.

Indexing an Attribute When You Create It

See: "Adding an Attribute" on page 6-22

Dropping an Index from an Attribute

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

You can use command line tools to perform the tasks described in the following sections:

- Adding and Modifying Attributes
- Indexing an Attribute by Using Command Line Tools

Adding and Modifying Attributes

See Also: "ldapmodify" on page A-11 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:

```
ldapmodify -h host -p port -f ldif_filename
```

The input file contains data similar to this:

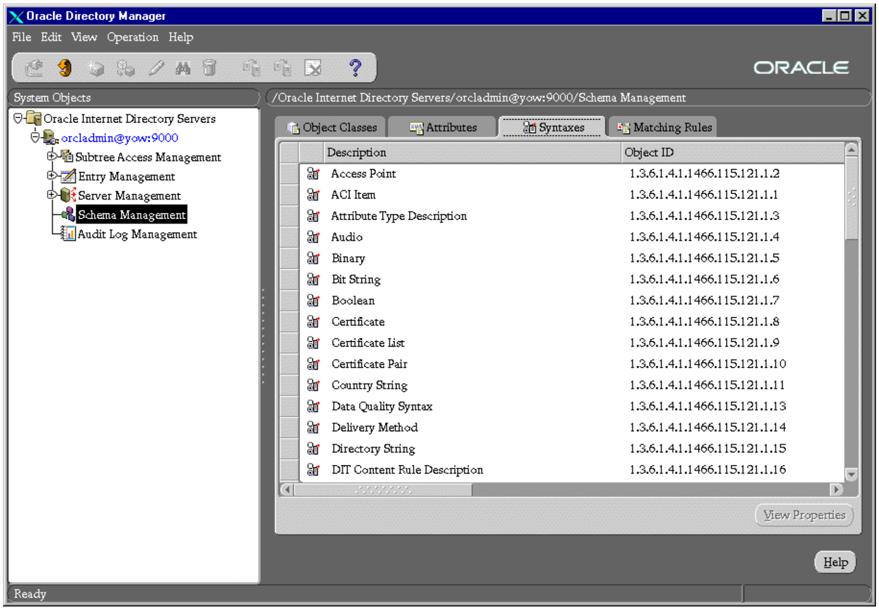
```
dn: cn=subschemasubentry
changetype: modify
add: attributetypes
attributetypes: ( 1.2.3.4.5 NAME 'myattr' SYNTAX
                 '1.3.6.1.4.1.1466.115.121.1.38' )
```

Finding a Syntax Object ID

You can find a given syntax Object ID by using either Oracle Directory Manager or the ldapsearch command line tool.

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:



Description	Object ID
Access Point	1.3.6.1.4.1.1466.115.121.1.2
ACI Item	1.3.6.1.4.1.1466.115.121.1.1
Attribute Type Description	1.3.6.1.4.1.1466.115.121.1.3
Audio	1.3.6.1.4.1.1466.115.121.1.4
Binary	1.3.6.1.4.1.1466.115.121.1.5
Bit String	1.3.6.1.4.1.1466.115.121.1.6
Boolean	1.3.6.1.4.1.1466.115.121.1.7
Certificate	1.3.6.1.4.1.1466.115.121.1.8
Certificate List	1.3.6.1.4.1.1466.115.121.1.9
Certificate Pair	1.3.6.1.4.1.1466.115.121.1.10
Country String	1.3.6.1.4.1.1466.115.121.1.11
Data Quality Syntax	1.3.6.1.4.1.1466.115.121.1.13
Delivery Method	1.3.6.1.4.1.1466.115.121.1.14
Directory String	1.3.6.1.4.1.1466.115.121.1.15
DIT Content Rule Description	1.3.6.1.4.1.1466.115.121.1.16

Viewing Syntaxes by Using by Using ldapsearch Use ldapsearch on the subentry `cn=subSchemaSubentry`.

> **See Also:** "ldapsearch" on page A-4

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 the recommended method is by using the Catalog Management tool.

These topics are discussed in the following sections:

- Indexing an Attribute for Which No Directory Data Exists
- Indexing an Attribute for Which Directory Data Exists

Indexing an Attribute for Which *No* Directory Data Exists

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" on page A-11

Indexing an Attribute for Which Directory Data Exists

Use the Catalog Management Tool to index an attribute for which data already exists and to drop an index from an attribute.

See: "Using the Catalog Management Tool" on page 4-15

7

Managing Directory Entries

This chapter covers topics in the following sections.

- Managing Entries by Using Oracle Directory Manager
- Managing Entries by Using Command Line Tools
- Managing Entries by Using Bulk Tools

See Also:

- Chapter 2 for an overview of directory entries, Directory Information Trees, Distinguished Names, and Relative Distinguished Names
- "Using Audit Log" on page 5-22 for instructions on how to search for entries in the audit log

Managing Entries by Using Oracle Directory Manager

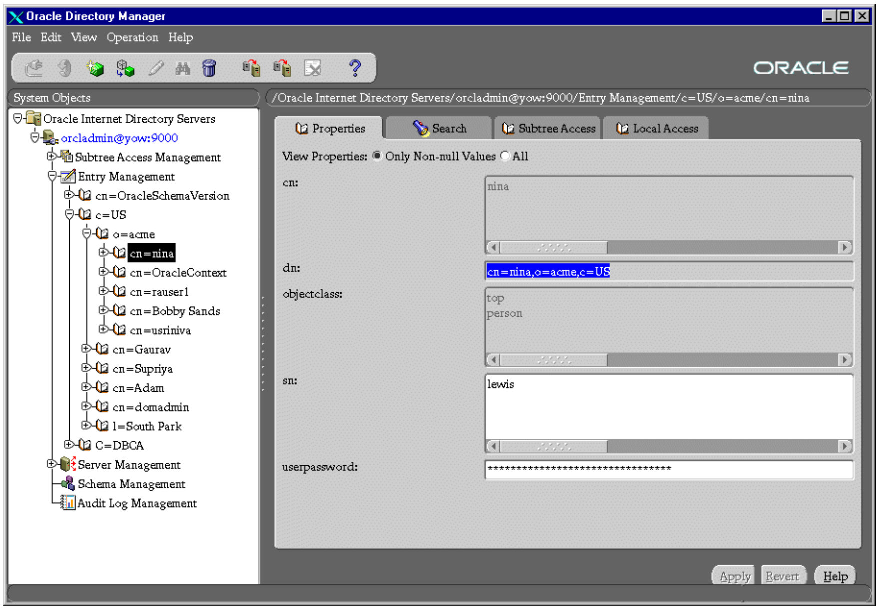
This section discusses the following tasks:

- Searching for Entries
- Searching for Audit Log Entries
- Viewing Directory Entry Attributes
- Adding Entries
- Modifying Entries

Searching for Entries

You can display all entries by using the navigator pane, or search for one or more specific entries by using Oracle Directory Manager's search feature.

To display an entry by using the navigator pane, expand Entry Management to display its subtree:



View Properties: ● Only Non-null Values ○ All

cn: nina

dn: cn=nina,o=acme,c=US

objectclass: top
person

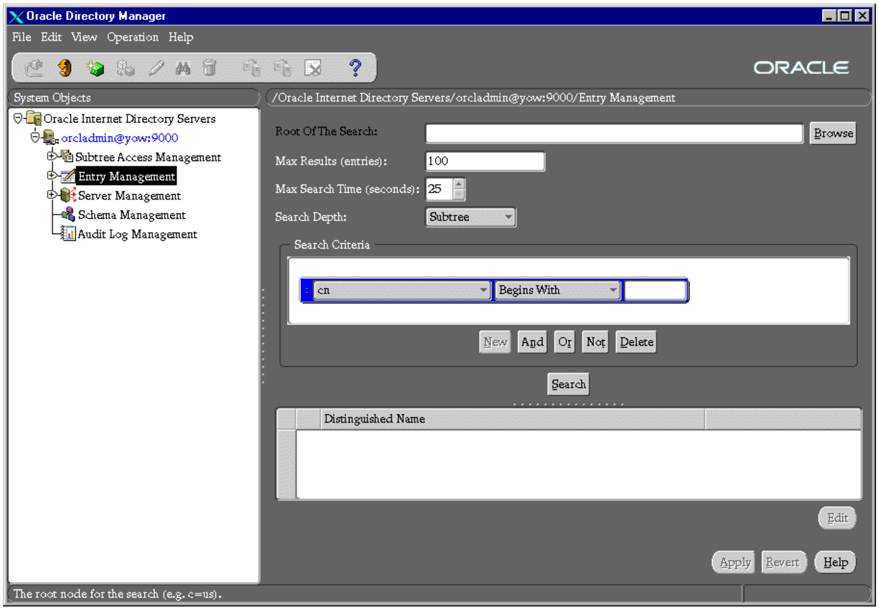
sn: lewis

userpassword: **********************************

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, select Entry Management. The Search fields appear in the right pane:



/Oracle Internet Directory Servers/orcladmin@yow:9000/Entry Management

Root Of The Search:

Max Results (entries): 100

Max Search Time (seconds): 25

Search Depth: Subtree

Search Criteria: cn Begins With

The root node for the search (e.g. c=us).

2. In the Root of the Search text box, enter the **DN** of the root of your search.

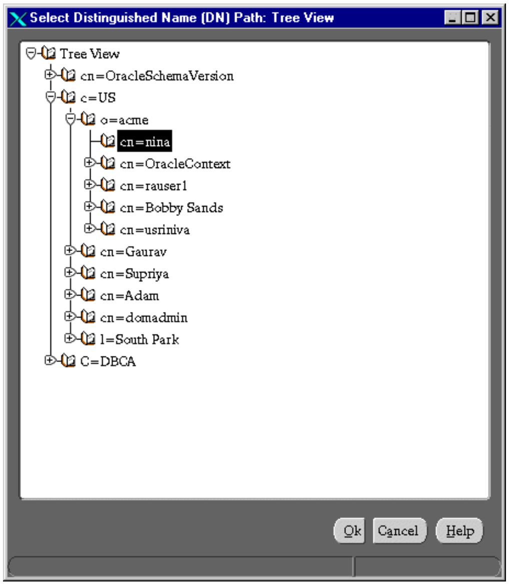
   For example, suppose you want to search for an employee who works in the Manufacturing division in the IMC organization in the Americas. The DN of the root of your search would be:

   ```
   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 tree. To do this:

a. Click the Browse button to the right of the Root of the Search text box. The Select Distinguished Name (DN) Path: Tree View dialog box appears:



b. Click the + 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 and click OK. The DN for the root of your search appears in the Root of the Search text box in the right pane.

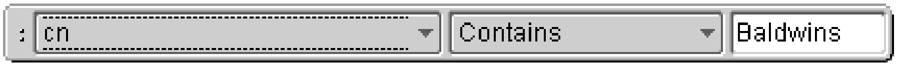
3. In the Max Results (entries) text box in the search window of Oracle Directory Manager, type the maximum number of entries you want your search to retrieve. The default is 100.

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.

5. In the Search Depth list, select the level to which you want to search. The options are:

- Base: To retrieve 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: To limit your search to all entries beginning one level down from the root of your search
- Subtree: To search entries within the entire subtree, including the root of your search

6. In the Search Criteria field, use the lists and text boxes on the search criteria bar to focus your search.



: cn Contains Baldwins

   a. In the menu at the left end of the search criteria bar, select an attribute of the entry for which you want to search.

   Note: 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. 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.

c. In the menu in the middle of the search criteria bar, select a filter. Options are:

Filter	Description
Begins With	To search by using only the first few characters of the attribute's value. For example, `cn Begins With Fr` retrieves all entries in which the first few letters of the `cn` attribute are `Fr`. These would include Frank, Fran, Frances, Franklin, etc.
Ends With	To search for an entry by using only the last few characters of the specified attribute's value.
Contains	To search 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, Winship, etc.
Exact Match	To search 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	To search 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	To search 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	To determine 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.

7. Beneath the Search Criteria field are five buttons described in Table 7–1. Use these buttons to further refine your search by enhancing the search criteria bar.

Table 7–1 Search Criteria Buttons

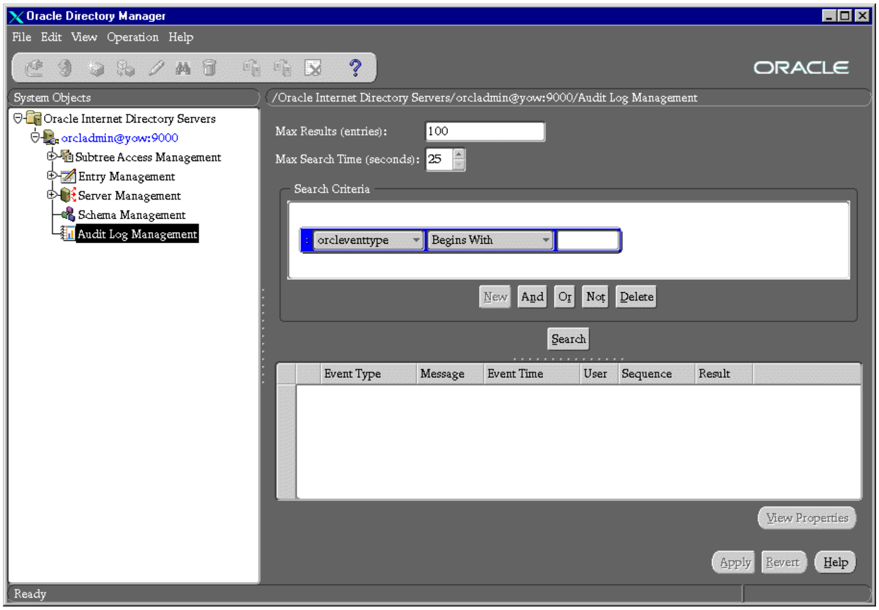
Button	Description
New	Creates a new search criteria bar in the Search Criteria field. This button is enabled only when the Search Criteria field is empty.
And	Creates another search criteria bar in the Search Criteria field. Matches all entries with one specified attribute with those that also have 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

8. Click Search. The results of your search appear in the Distinguished Name window of the right pane.

Searching for Audit Log Entries

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

To use Oracle Directory Manager to view audit log entries, in the navigator pane, select Audit Log Management. The corresponding right pane appears.


/Oracle Internet Directory Servers/orcladmin@yow:9000/Audit Log Management

Max Results (entries): 100

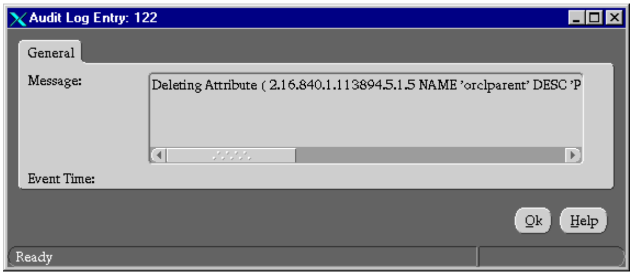
Max Search Time (seconds): 25

Search Criteria: orcleventtype Begins With

Use this pane to search for particular types of entries in the audit log.

The results of the search appear in the lower box. You can scroll both horizontally and vertically to see all the information.

To view the properties of a particular audit log entry, select it in the lower box and click View Properties. The Audit Log Entry dialog box displays the properties for the audit log entry you selected:



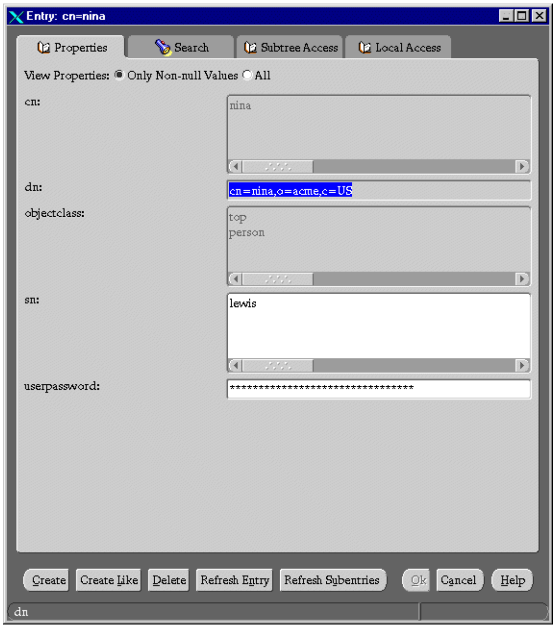
Message: Deleting Attribute (2.16.840.1.113894.5.1.5 NAME 'orclparent' DESC 'P

Event Time:

See Also: "Searching for Entries" on page 7-2

Viewing Directory Entry Attributes

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:



View Properties: ● Only Non-null Values ○ All

cn: nina

dn: cn=nina,o=acme,c=US

objectclass: top
person

sn: lewis

userpassword: **********************************

dn

Some attributes may 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. The Directory: Entry Management dialog box appears. Follow the steps mentioned in "Searching for Entries" on page 7-2.

Adding Entries

You can use Oracle Directory Manager to add entries as described in the following sections:

- Adding a New Entry
- Adding an Entry by Copying an Existing Entry
- Adding Group Entries

Note: This release of Oracle Internet Directory does not support the adding of JPEG images by using Oracle Directory Manager. You may add a JPEG image by using the ldapadd command. For more information, see "Example: Adding a User Entry by Using ldapadd" on page 7-24.

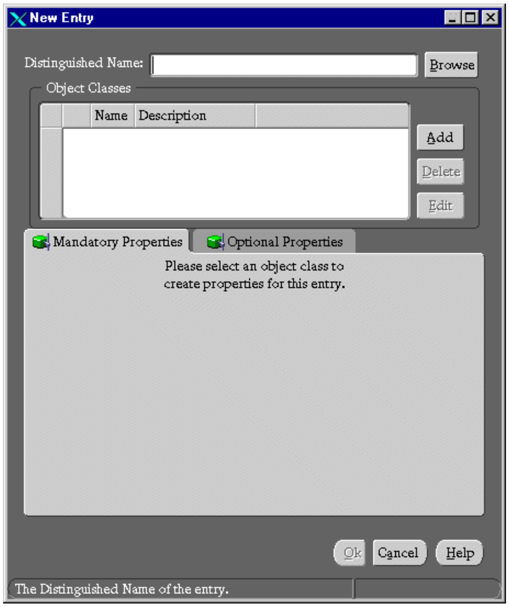
Adding a New Entry

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.

See Also: Access Control and Authorization on page 2-25 and Chapter 9 for information on access privileges

To add a new entry:

1. Either click the Create button on the tool bar, or select Create Entry from the Operation menu. The New Entry dialog box appears:

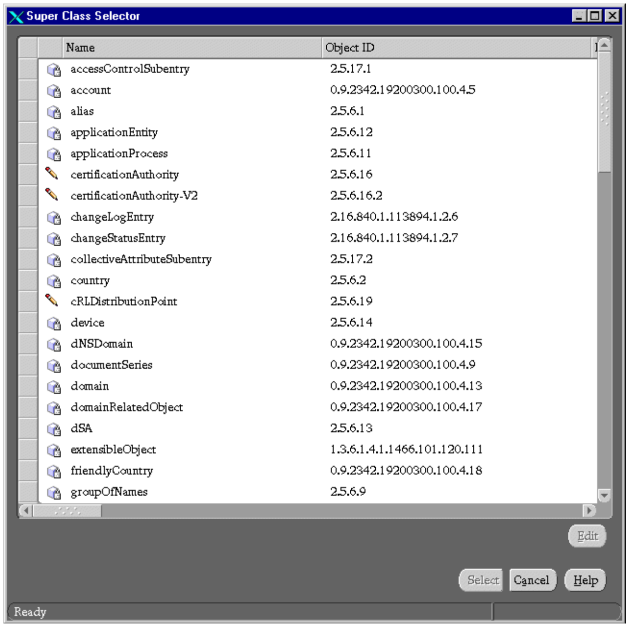


Please select an object class to create properties for this entry.

The Distinguished Name of the entry.

2. In the Distinguished Name field, type the full DN. You may also click Browse to locate the DN of the parent for the entry you want to add, then type the RDN for your new entry to the left of that parent DN.

3. To specify the **object classes** you want to use for the new entry, click the Add button to the right of the Object Classes window. The Super Class Selector dialog box appears:



Name	Object ID
accessControlSubentry	2.5.17.1
account	0.9.2342.19200300.100.4.5
alias	2.5.6.1
applicationEntity	2.5.6.12
applicationProcess	2.5.6.11
certificationAuthority	2.5.6.16
certificationAuthority-V2	2.5.6.16.2
changeLogEntry	2.16.840.1.113894.1.2.6
changeStatusEntry	2.16.840.1.113894.1.2.7
collectiveAttributeSubentry	2.5.17.2
country	2.5.6.2
cRLDistributionPoint	2.5.6.19
device	2.5.6.14
dNSDomain	0.9.2342.19200300.100.4.15
documentSeries	0.9.2342.19200300.100.4.9
domain	0.9.2342.19200300.100.4.13
domainRelatedObject	0.9.2342.19200300.100.4.17
dSA	2.5.6.13
extensibleObject	1.3.6.1.4.1.1466.101.120.111
friendlyCountry	0.9.2342.19200300.100.4.18
groupOfNames	2.5.6.9

4. 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.

5. When you have selected the object classes and provided values for the appropriate attributes, click OK.

Adding an Entry by Copying an Existing Entry

You can use Oracle Directory Manager to create a new entry by copying from an existing entry and changing its DN. 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.

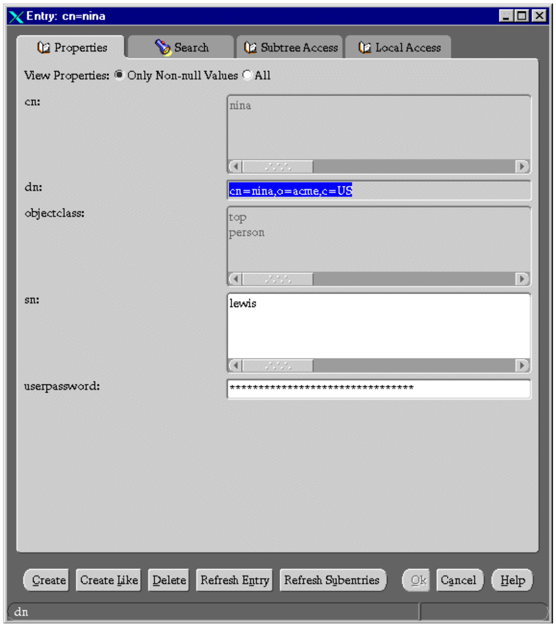
To add an entry by copying an existing entry:

1. When you click Entry Management in the navigator pane, the Search pane appears. Use it to search for an entry that you want to use as a template.

   See Also:

   - "Managing Entries by Using Oracle Directory Manager" on page 7-2 for instructions on using the search pane
   - The online help

2. Double-click an entry from those retrieved. The Entry dialog box for that entry appears:



View Properties: ● Only Non-null Values ○ All

cn: nina

dn: cn=nina,o=acme,c=US

objectclass: top
person

sn: lewis

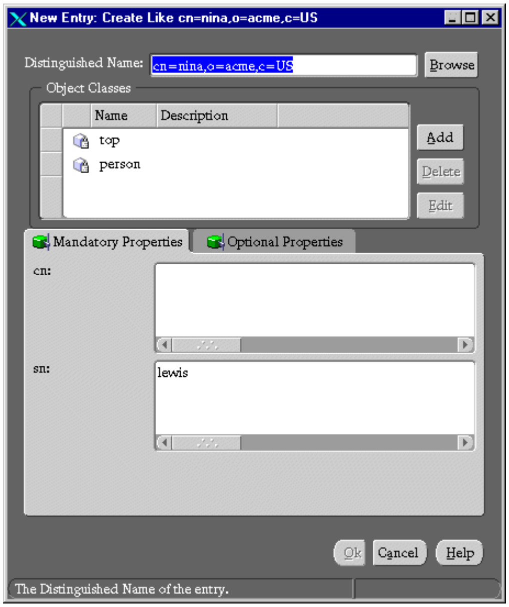
userpassword: ********************************

dn

This entry will serve as your template in the Create Like pane.

3. Click the Create Like button in the Entry dialog box.

A New Entry: Create Like window appears:



4. Change critical fields to tailor this particular 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 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 have to change the Henri Latour value in the common name attribute to Henri Latrobe, and any other attributes that must be unique, such as employee number and telephone number.

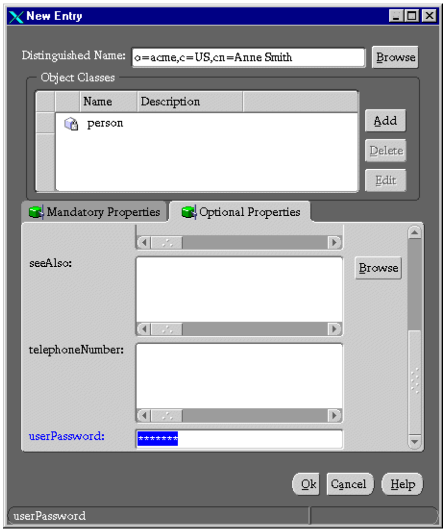
5. Click OK to save your changes.

> **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 > *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, to the left of that parent DN.
5. Click the Add button to the right of the Object Classes window. 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. The Optional Properties tab in the New Entry dialog box now looks something like the following:



New Entry

Distinguished Name: o=acme,c=US,cn=Anne Smith

Object Classes

Name	Description
person	

seeAlso:

telephoneNumber:

userPassword: *******

userPassword

Adding Group Entries

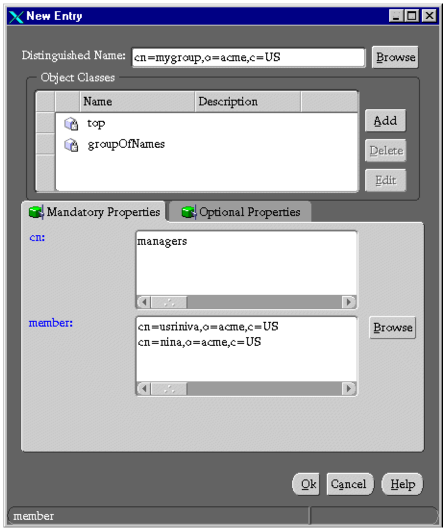
A group entry is one that contains a list of entries, for example, an e-mail list. You associate it with either the `groupOfNames` or `groupOfUniqueNames` object class, which has the object class `orclPrivilegeGroup` as a subclass.

You determine membership in the group the group by adding DNs to the multi-valued 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. Either click the Create button on the tool bar, or select Create Entry from the Operation menu. The New Entry dialog box appears.
2. 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 your new entry to the left of that parent DN.
3. To specify the object classes you want to use for the new entry, click the Add button to the right of the Object Classes window. The Super Class Selector dialog box appears.
4. 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 window of the New Entry dialog box.

5. In the same way, click the Add button to the right of the Object Classes window and, from the Super Class Selector dialog box, select the `groupOfNames` or `groupOfUniqueNames` object class. Click the Select button. The object class you selected appears in the Object Classes window of the New Entry dialog box.



Distinguished Name: cn=mygroup,o=acme,c=US

Object Classes

Name	Description
top	
groupOfNames	

cn: managers

member: cn=usriniva,o=acme,c=US
cn=nina,o=acme,c=US

member

6. Enter the mandatory and optional attributes for your group entry.

   Note the Browse button next to the member window. Clicking Browse displays the Directory: Entry Management dialog box. Use this dialog box to search for a particular entry you want to add to the list. In the Distinguished Name window of the Directory: Entry Management dialog box, select the entry, then click the OK button. This returns you to the New Entry dialog box. The entry you just selected is added to the list in the members window.

7. After you have completed the attribute fields, click OK.

   See Also: "Privilege Groups" on page 9-4 for instructions on setting access control policies for group entries

Modifying Entries

Oracle Directory Manager is governed by standard LDAP conventions, including the following:

- You cannot change object classes that are used by an entry once you have assigned object classes to that entry and populated its attributes with data.

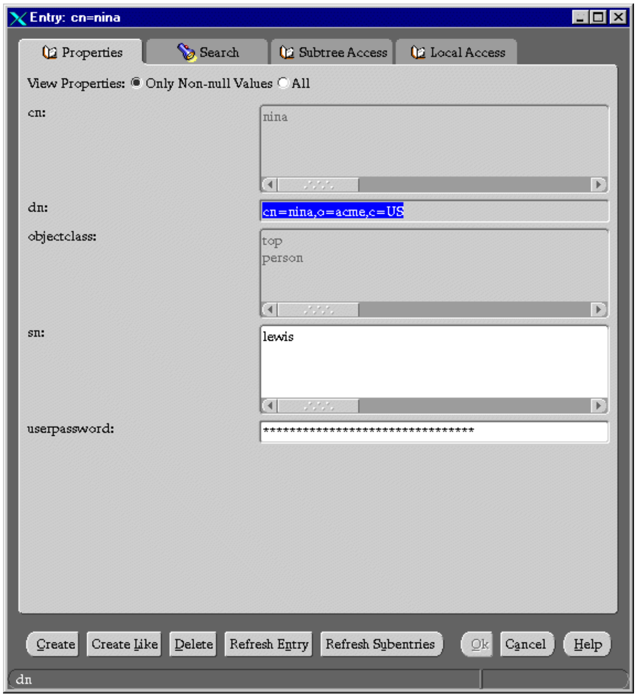
  For example, if you configure an entry to use object classes `Person` and `Organizational Role`, you cannot later add another object class to this entry.

- You may *not* 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.
2. In the Distinguished Name window of the Search pane, select the entry you want to modify.

3. Click Edit. The Entry dialog box appears.



Entry: cn=nina

View Properties: ● Only Non-null Values ○ All

cn: nina

dn: cn=nina,o=acme,c=US

objectclass: top
person

sn: lewis

userpassword: ********************************

4. In the Entry dialog box, modify the values of any editable attributes, then click OK.

Example: Modifying a User Entry by 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 7-17.

1. Perform a search for the Anne Smith entry.
2. In the Distinguished Name window of the Search pane, 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 by Using Command Line Tools

The following table summarizes some of the more common entry management tasks and the corresponding tool(s) for each one.

Task	Tool								
	ldapsearch	ldapbind	ldapadd	ldapaddmt	ldapmodify	ldapmodifymt	ldapdelete	ldapcompare	ldapmoddn
Add a single entry			X						
Add multiple entries concurrently				X					
Add new configuration set entries			X						
Compare attribute values you specify with those in a directory entry								X	
Configure a server with an input file			X						
Delete an entry					X		X		
Modify attribute values					X				
Modify configuration set entries					X				
Modify DN or RDN of an entry					X				X
Modify several entries concurrently						X			
Move an entry or subtree under a new parent									X
Rename a subtree									X
Rename an entry									X
Search for an entry	X								
Verify that you can connect a client to a server		X							

The following table lists each of the command line tools, and tells you where to find syntax and usage notes for each one.

Tool	Information
ldapsearch	"ldapsearch" on page A-4
ldapbind	"ldapbind" on page A-6
ldapadd	"ldapadd" on page A-7
ldapaddmt	"ldapaddmt" on page A-9
ldapmodify	"ldapmodify" on page A-11
ldapmodifymt	"ldapmodifymt" on page A-14
ldapdelete	"ldapdelete" on page A-16
ldapcompare	"ldapcompare" on page A-17
ldapmoddn	"ldapmoddn" on page A-18

Example: Adding a User Entry by Using ldapadd

In this example, we add the user entry for Audrey found in the file `entry.ldif`:

```
ldapadd -p 389 -b -f entry.ldif
```

This LDIF file contains the `cn`, `sn`, `jpegPhoto`, and `userpassword` attributes. For the `jpegPhoto` attribute, it specifies the path and file name of the corresponding JPEG image.

```
dn: cn=audrey, c=us
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: audrey
sn: hepburn
jpegPhoto: /photo/audrey.jpg
userpassword: welcome
```

Note that, in this user entry, the `jpegPhoto` attribute specifies the path and file name of the JPEG image you want to include as an entry attribute.

Example: Modifying a User Entry by Using ldapmodify

In this example, we change 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 file `entry.ldif`.

```
ldapmodify -p 389 -b  -f entry.ldif
```

The LDIF file in this example contains the following:

```
dn: cn=audrey, c=us
changetype:modify
replace: userpassword
userpassword:audreyspassword
```

Managing Entries by Using Bulk Tools

This section lists and describes some of the more common tasks you perform with bulk tools. For an overview of these tools, see "Using Bulk Tools" on page 4-14.

This section discusses administrative tasks in the following sections:

- Importing an LDIF File by Using bulkload
- Converting Directory Data to LDIF
- Modifying a Large Number of Entries
- Deleting a Large Number of Entries

Importing an LDIF File by Using bulkload

To import an LDIF file, you use the bulkload utility. The steps to process an LDIF file through bulkload are briefly summarized immediately below and are explained in detail later in this section.

- Step 1: Back up the Oracle server
- Step 2: Find out the Oracle Internet Directory password
- Step 3: Check input for schema and data consistency violations
- Step 4: Generate the input files for SQL*Loader
- Step 5: Load the input files

Step 1: Back up the Oracle server

Before you import the file, back up the Oracle server as a safety precaution.

> **See Also:** *Oracle8i Backup and Recovery Guide*

Step 2: Find out the Oracle Internet Directory password

To use bulkload and the other shell script tools which 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 4-16

Step 3: Check input for schema and data consistency violations

On a Solaris computer, the file `bulkload.sh` usually resides in `$ORACLE_HOME/ldap/bin`. On a Windows NT computer, 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`.

Step 4: Generate the input files for SQL*Loader

After you have fixed any errors in the input file, re-run bulkload with the `-generate` option. 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.

Step 5: Load the input files

After you have generated the input files, re-run 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 `$ORACLE_HOME/ldap/log/ *.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: "ldifwrite" on page A-21

Modifying a Large Number of Entries

The bulkmodify utility enables you to modify a large number of existing entries in an efficient way.

See: "bulkmodify" on page A-23

Deleting a Large Number of Entries

The bulkdelete utility enables you to delete an entire subtree efficiently.

See: "bulkdelete" on page A-25

8

Managing Secure Sockets Layer (SSL)

If you use Secure Sockets Layer (SSL), you may also configure strong authentication, data integrity, and data privacy. This section discusses these topics in the following sections:

- Supported Cipher Suites
- SSL Client Scenarios
- Configuring SSL Parameters
- Issues Specific to This Release of Oracle Internet Directory

See Also: "Security" on page 2-20 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 8–1 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 either anonymously or by using simple authentication.

When both a client and server authenticate themselves to each other, SSL derives the identity information it requires from the certificate.

Configuring SSL Parameters

During start-up of a 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.

See Also:

- "Managing Server Configuration Set Entries" on page 5-2 for instructions on how to set these parameters
- Table F–1 on page F-5 for a description of these parameters

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 is the model from which all other configuration sets are drafted.

Configuring SSL Parameters by Using Oracle Directory Manager

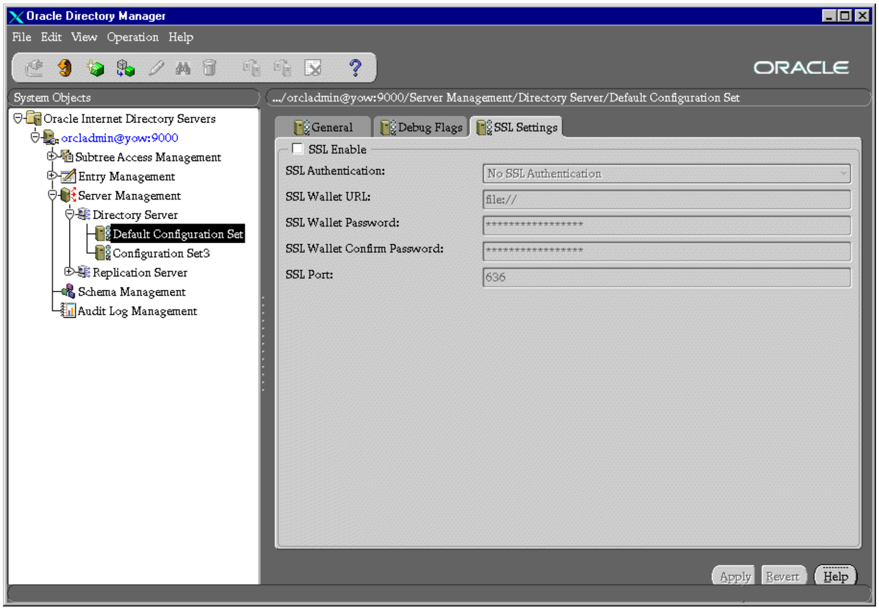
You can examine and modify the values for the SSL configuration parameters in each configuration set entry that you have created and in each server instance that is currently running.

> **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 > *server you want to explore* > Server Management.
1. Expand either Directory Server or Replication Server, as appropriate. The numbered configuration sets are listed beneath your selection.
2. Select the configuration set that you want to examine. The group of tab pages for that configuration set entry appear in the right pane.

3. Select the SSL Settings tab:



□ SSL Enable

SSL Authentication: No SSL Authentication

SSL Wallet URL: file://

SSL Wallet Password: *****************

SSL Wallet Confirm Password: *****************

SSL Port: 636

You can change the parameters in this tab page and save them. The fields in this tab page are described in Table 5–2 on page 5-8.

> **See Also:** "Managing Server Configuration Set Entries by Using Oracle Directory Manager" on page 5-4 for information on changing parameters in a configuration set entry

Configuring SSL Parameters by Using Command Line Tools

See: "Managing Server Configuration Set Entries by Using Command Line Tools" on page 5-11.

Issues Specific to This Release of Oracle Internet Directory

In this release, the replication server cannot communicate with SSL-enabled servers.

If you intend to support both SSL and non-SSL clients on the same host, you need to configure two distinct server instances.

See Also: Chapter 5 for instructions on how to configure server instances

9

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.

See Also:

- The conceptual material found in the section "Access Control and Authorization" on page 2-25 before you begin implementing and administering access control policies
- Appendix E for information about the format or syntax of Access Control Items (ACIs)

This chapter covers topics in the following sections:

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 any other tool that supports the standard LDAP modify operation.

This section discusses topics in the following subsections:

- Access Control Management Constructs
- Access Control Information Components
- ACL Evaluation
- Modifying Existing ACPs and their ACI Directives
- Adding an ACP and Creating Access Items
- Managing ACPs: An Example
- Granting Entry-Level Access
- Managing Access Control: Examples

Access Control Management Constructs

This section discusses topics in the following subsections:

- orclACI
- Access Control Policy Points
- orclEntryLevelACI
- Privilege Groups

orclACI

`orclACI` is a user modifiable optional attribute that represents **Access Control List** (ACL) policy information for a subtree starting with the entry where that subtree is defined. Any entry in the directory can contain values for this attribute. Access to this attribute itself can be controlled the same way access to any other attribute is controlled.

Access Control Policy Points

Access Control Policy Points (ACPs) are entries in which the `orclACI` attribute value is set. The `orclACI` attribute value represents the access policies inherited by a subtree of entries starting with the ACP as the root of the subtree.

When a hierarchy of multiple ACPs exists in a directory subtree, the subordinate entries in that subtree inherit the access policies from all of the ACPs that are superior to the entry. 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 root entry for the HR department, and the Benefits, Payroll, and Insurance groups are entries within the HR department, any entry within those groups inherits the access rights specified in that HR root entry.

When there are conflicting policies represented among a hierarchy of ACPs, the directory applies well defined precedence rules while evaluating the aggregate policy.

See Also: "ACL Evaluation" on page 9-10.

orclEntryLevelACI

The `orclACI` attribute contains ACL directives that are prescriptive in nature, that is, the directives apply to all entries in the subtree below the ACP where this attribute is defined. In addition, it is often convenient to maintain ACL directives specific to a single entry within that entry. Oracle Internet Directory enables you to do this through a user modifiable operational attribute called `orclEntryLevelACI`. Any directory entry can optionally carry a value for this attribute. This is accomplished in the Oracle Internet Directory base schema by extending 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.

It is possible to represent ACL directives specific to a single entry in the `orclACI` attribute. However, for administrative convenience and performance advantages, Oracle Corporation recommends using `orclEntryLevelACI` in such scenarios. 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`.

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 DN, Oracle Internet Directory computes his or her 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. One can set access control policies such that they are applicable to the members of group1, group2, and group3.

```
dn: cn=group1, c=us
cn: group1
objectclass: top
objectclass: groupofUniquenames
objectclass: orclprivilegegroup
uniquemember:  cn=john smith, c=us
uniquemember:  cn=joe smith, c=us
uniquemember:  cn=bill smith, c=us
```

```
dn: cn=group2, c=us
cn: group2
objectclass: top
objectclass: groupofUniquenames
objectclass: orclprivilegegroup
uniquemember:  cn=john smith, c=us
uniquemember:  cn=joe smith, c=us
uniquemember:  cn=bill smith, c=us
```

```
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
```

```
dn: cn=group4, c=us
cn: group4
objectclass: top
objectclass: groupofUniquenames
uniquemember:  cn=john smith, c=uk
uniquemember:  cn=joe smith, c=uk
uniquemember cn=bill 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, semantically, an ACI is a tuple consisting of three components described in the following sections:

- Object: To What Are You Granting Access?
- Subject: To Whom Are You Granting Access?
- Operations: What Access Are You 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 root. 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 not required nor syntactically allowed.

In the orclEntryLevelACI attribute, the entry DN component of the object of the ACL is implicitly that of entry itself. For example, if dc=oracle,dc=com has an entry level ACI associated with it, the entry governed by its ACI is exactly: dc=oracle, 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 select any entry, and is convenient shorthand for the equivalent `dn=.*` selector.

Attributes within an entry are selected by including a comma-separated list of attribute names in the object selector.

`attr=(`*`attribute_list`*`)`

Attributes within an entry are rejected 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:

- "Managing Access Control: Examples" on page 9-44 for examples using command line tools
- Appendix E for information about the format or syntax of ACI

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 five modes:

- `Anonymous`: No authentication (default mode)
- `Simple`: Simple password-based authentication
- `SSLNoauth`: For SSL-based clients with either anonymous or simple password based authentication
- `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.

The BindMode is optional in subject specification. When specified, the mode should match the mode specified in the ACI for the directive to be applicable.

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 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`*`)`

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.

Access Level	Description
None	No access right
Add	Right to add entries under a target directory entry
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 to use an entry DN as the base DN in ldapsearch operation.
Compare	Right to perform compare operation on the attribute
Delete	Right to delete the target entry
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 modify (add / delete) its own value in the attribute. For example, right to add / delete oneself in a list of DNs group entry attribute.
Write	Right to modify / add / delete the attributes of an entry. To delete an entry the user must have delete access to the entry.

Note that some access permissions are associated with entries and others with attributes.

Permissions for Entries:	**Permissions for Attributes:**
Browse / nobrowse	Compare / nocompare
Add / noadd	Search / nosearch
Delete / nodelete	Read / noread
None	Selfwrite / noselfwrite
	Write / nowrite
	None

The entry level access directives are distinguished by the keyword ENTRY in the object component.

ACL Evaluation

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. This process is described here.

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 either if access to the entry itself is denied or if any of the attributes reach the resolved with denial state. In this case the operation would fail.

For a search operation, the evaluation continues until all the attributes reach the resolved state. Attributes that are resolved with denial are not returned.

The remainder of this section covers the topics in the following subsections:

- ACL Evaluation Precedence Rules
- Assigning More Than One ACI to the Same Object
- Granting Exclusionary Access to Objects
- ACL Evaluation For Groups

ACL Evaluation Precedence Rules

An LDAP operation requires the BindDN—that is, the 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 fully resolves:

- 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:
  - UNION (all the granted permissions in the matching by clause items)

    and
  - UNION (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. With a filter. For example:

   ```
   access to attr=(salary) filter=(salary >10000)
       by group1 (read)
   ```

2. 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 the first ACI that happens to be returned wins. 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 returned first, it wins, and 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 excluding that one, you must be sure 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 it is specified (See the section "ACL Evaluation Precedence Rules" on page 9-11). But this means that anyone in group2 who tries to access the userpassword attribute could not be 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’s 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, typically, the ACL evaluation for the attribute (or entry) is considered Resolved with Denial. But, if the user of the session (bindDN) is known to be a member of a group object, the evaluation will continue 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, such grants have higher precedence than any denials below 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 attributes that are being modified
Remove an object	Delete access to the object being removed

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:** After installing Oracle Internet Directory, you should immediately modify the Default ACP—reducing the totally open default set of permissions to allow total access only to system administrators.

See Also: Appendix A for a description of command line tools

Instructions for managing access control by using Oracle Directory Manager are contained in the following sections:

- Modifying Existing ACPs and their ACI Directives
- Adding an ACP and Creating Access Items
- Managing ACPs: An Example
- Granting Entry-Level Access

Modifying Existing ACPs and their ACI Directives

ACPs are entries that contain 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.

This section discusses topics in the following subsections:

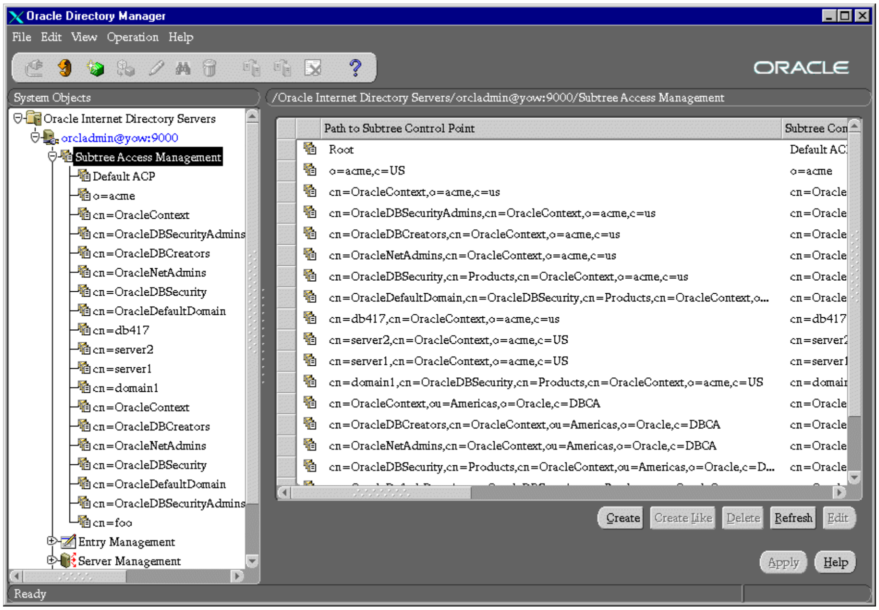
- Viewing an ACP
- Adding Structural Access Items to an Existing ACP
- Adding Content Access Items to an Existing ACP
- Modifying Structural Access Items of an Existing ACP
- Modifying Content Access Items of an Existing ACP

Viewing an ACP

To view an ACP:

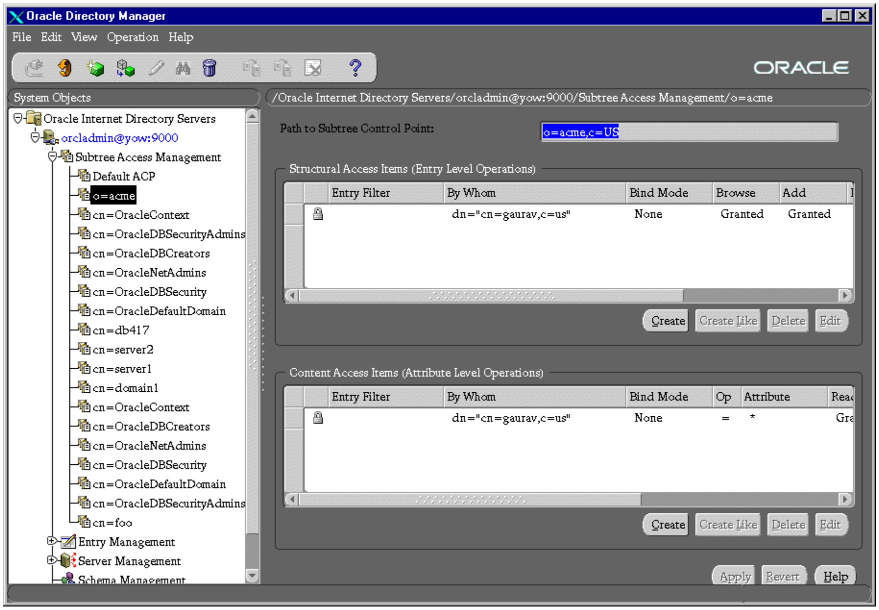
1. Start Oracle Directory Manager and connect to an Oracle Internet Directory server.

2. In the navigator pane, select and expand Subtree Access Management. All of the defined Access Control Policy Points (ACPs) appear in a list below Subtree Access Management in the navigator pane and in the right pane:



Path to Subtree Control Point	Subtree Con
Root	Default AC
o=acme,c=US	o=acme
cn=OracleContext,o=acme,c=us	cn=Oracle
cn=OracleDBSecurityAdmins,cn=OracleContext,o=acme,c=us	cn=Oracle
cn=OracleDBCreators,cn=OracleContext,o=acme,c=us	cn=Oracle
cn=OracleNetAdmins,cn=OracleContext,o=acme,c=us	cn=Oracle
cn=OracleDBSecurity,cn=Products,cn=OracleContext,o=acme,c=us	cn=Oracle
cn=OracleDefaultDomain,cn=OracleDBSecurity,cn=Products,cn=OracleContext,o...	cn=Oracle
cn=db417,cn=OracleContext,o=acme,c=us	cn=db417
cn=server2,cn=OracleContext,o=acme,c=US	cn=server2
cn=server1,cn=OracleContext,o=acme,c=US	cn=server1
cn=domain1,cn=OracleDBSecurity,cn=Products,cn=OracleContext,o=acme,c=US	cn=domair
cn=OracleContext,ou=Americas,o=Oracle,c=DBCA	cn=Oracle
cn=OracleDBCreators,cn=OracleContext,ou=Americas,o=Oracle,c=DBCA	cn=Oracle
cn=OracleNetAdmins,cn=OracleContext,ou=Americas,o=Oracle,c=DBCA	cn=Oracle
cn=OracleDBSecurity,cn=Products,cn=OracleContext,ou=Americas,o=Oracle,c=D...	cn=Oracle

3. Select an ACP under Subtree Access Management in the navigator pane to display its information in the right pane:



/Oracle Internet Directory Servers/orcladmin@yow:9000/Subtree Access Management/o=acme

Path to Subtree Control Point: o=acme,c=US

Structural Access Items (Entry Level Operations)

	Entry Filter	By Whom	Bind Mode	Browse	Add
🔒		dn="cn=gaurav,c=us"	None	Granted	Granted

Content Access Items (Attribute Level Operations)

	Entry Filter	By Whom	Bind Mode	Op	Attribute	Rea
🔒		dn="cn=gaurav,c=us"	None	=	*	Gra

You can alternatively double-click an ACP in the right pane to display the data in its own window.

The three fields in the Subtree Access Management pane are:

Field	Description
Path to the Subtree Access Control Point	The Path to 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 the Subtree Access Control Point field. If you are creating a new ACP, you must enter the path to it here.
Structural Access Items (Entry Level Operations)	Structural access items determine access to entries. Items listed in the Structural Access Items window 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 For instructions on how to modify structural access items, see "Modifying Structural Access Items of an Existing ACP" on page 9-28.

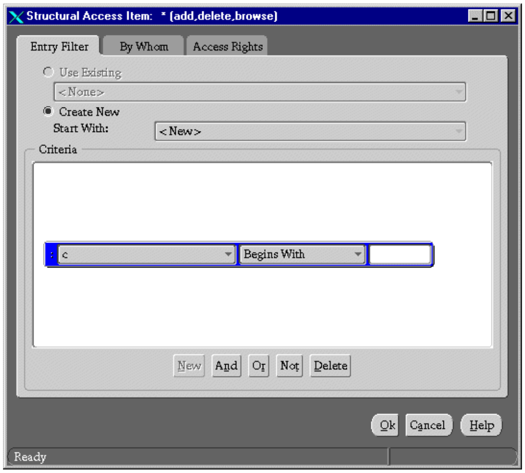
Field	Description
Content Access Items (Attribute Level Operations)	Items listed in the Content Access Items window are related to attributes within the entry or entries identified in the Entry Filter field. Fields in this window include By Whom, Bind Mode, as well as: ■ Op: The matching operation to 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 For instructions on how to modify content access items, see "Modifying Content Access Items of an Existing ACP" on page 9-32

Note: Scroll horizontally to read all the access permissions.

Adding Structural Access Items to an Existing ACP

To add a structural access item to an existing ACP:

1. In the navigator pane, expand Subtree Access Management. All of the defined Access Control Policy Points (ACPs) appear in a list below Subtree Access Management in the navigator pane. They also appear in the right pane.
2. Under Subtree Access Management in the navigator pane, select an ACP to display its information in the right pane.
3. Just below the Structural Access Items window, click the Create button. A Structural Access Items dialog box displays three tabs: Entry Filter, By Whom, and Access Rights:



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 secretary, or for all those whose title is manager and whose organization unit is Americas.

In the Criteria window of the Entry Filters tab, 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:

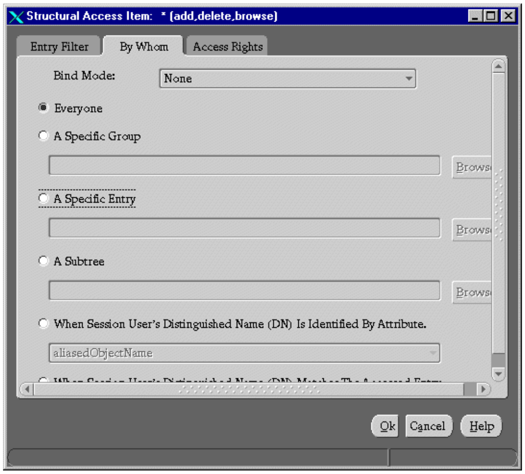
a. In the menu at the left end of the bar, select an attribute.

b. In the menu in the middle of the bar, select one of the following filter options:

Filter	Description
Begins With	To search by using only the first few characters of the attribute's value
Ends With	To search for an entry by using only the last few characters of the specified attribute's value
Contains	To search for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	To search for an entry whose specified attribute is the same as the value you enter
Greater or Equal	To search for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter
Less or Equal	To search for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter
Present	To determine 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 that 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 selected.

5. Select the By Whom tab to define the subject of the ACI:



Structural Access Item: * (add,delete,browse)

Bind Mode: None

● Everyone

○ A Specific Group

○ A Specific Entry

○ A Subtree

○ When Session User's Distinguished Name (DN) Is Identified By Attribute.

aliasedObjectName

In the By Whom tab page, 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:

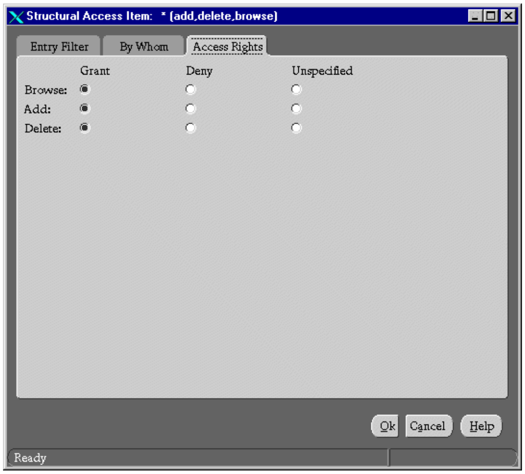
Bind Mode	Description
None	No authentication
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption / decryption is used.
SSL One Way	Only the directory server authenticates itself to the client. 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. Both the client and server send 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.

Also in the By Whom tab page, 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 (dn)	A previously defined group name
A specific entry (dn)	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

In the By Whom tab page, click OK.

6. Select the Access Rights tab page:



Structural Access Item: * (add,delete,browse)

	Grant	Deny	Unspecified
Browse:	●	○	○
Add:	●	○	○
Delete:	●	○	○

In the Access Rights tab, select the appropriate radio buttons to specify the kinds of rights you want to grant: Browse, Add, or Delete.

In the Access Rights tab page, click OK to close the Structural Access Items dialog box and return to the main Oracle Directory Manager dialog box. The structural ACI you just set is listed in the Structural Access Items window of the main Oracle Directory Manager dialog box.

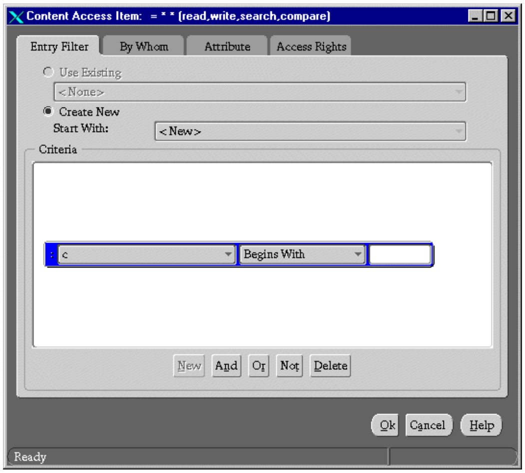
7. In the main Oracle Directory Manager dialog box, click Apply to send the data you have entered to the directory server.

Note: You must click Apply in the main Oracle Directory Manager dialog box in order 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.

Adding Content Access Items to an Existing ACP

To add a content access item to an existing ACP:

1. In the navigator pane, expand Subtree Access Management. All of the defined Access Control Policy Points (ACPs) appear in a list below Subtree Access Management in the navigator pane. They also appear in the right pane.
2. Select an ACP under Subtree Access Management in the navigator pane 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.
3. In the Content Access Items window of the right pane, select the Content Access Item you want to modify.
4. Just below the Content Access Item window, click the Create button. The Content Access Items dialog box appears:



Content Access Item: = * * (read,write,search,compare)

○ Use Existing

<None>

● Create New

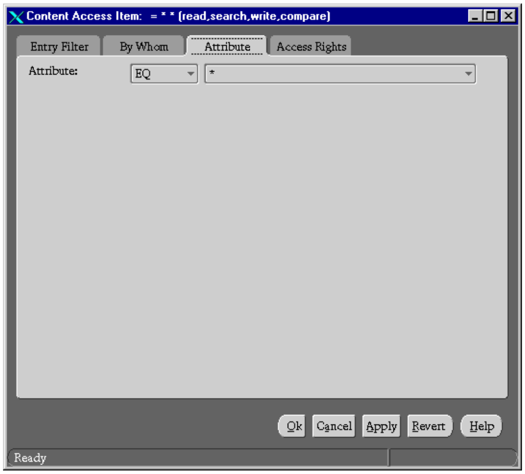
Start With: <New>

Criteria: c | Begins With |

This dialog box is similar to the Structural Access Items dialog box, but it has four tabs: Entry Filter, By Whom, Attribute, and Access Rights.

5. In the Entry Filter tab page, specify the items (if applicable) as described in the section "Adding Content Access Items to an Existing ACP" on page 9-26.
6. Select the By Whom tab and specify the items as described in the section "Adding Content Access Items to an Existing ACP" on page 9-26.

7. Select the Attribute tab:



Content Access Item: = * * (read,search,write,compare)

Attribute: EQ *

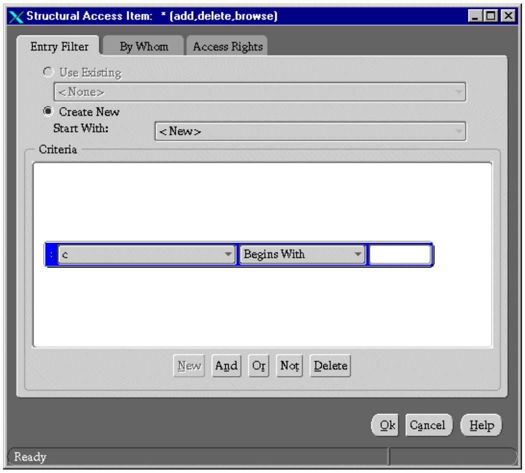
8. From the right menu in the Attribute Tab Page, select the attribute to which you want to grant or deny access.
9. In the left menu in the Attribute Tab Page, select the matching operation to be performed against the attribute. Choices are EQ (Equal (=)) and NEQ (Not Equal (!=)).
10. Select the Access Rights tab and specify the items as described in the section "Adding Content Access Items to an Existing ACP" on page 9-26.
11. Click OK.
12. In the main Oracle Directory Manager dialog box, click Apply.

Note: You must click Apply in the main Oracle Directory Manager dialog box in order 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.

Modifying Structural Access Items of an Existing ACP

To modify a structural access item:

1. In the navigator pane, expand Subtree Access Management. All of the defined Access Control Policy Points (ACPs) appear in a list below Subtree Access Management in the navigator pane. They also appear in the right pane.
2. Under Subtree Access Management in the navigator pane, select an ACP to display its information in the right pane.
3. Just below the Structural Access Items window, click the Edit button. The Structural Access Item dialog box appears:



Structural Access Item: * (add,delete,browse)

○ Use Existing

<None>

● Create New

Start With: <New>

Criteria: c Begins With

Ready

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 secretary, or for all those whose title is manager and whose organization unit is Americas.

In the Criteria window of the Entry Filters tab, 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:

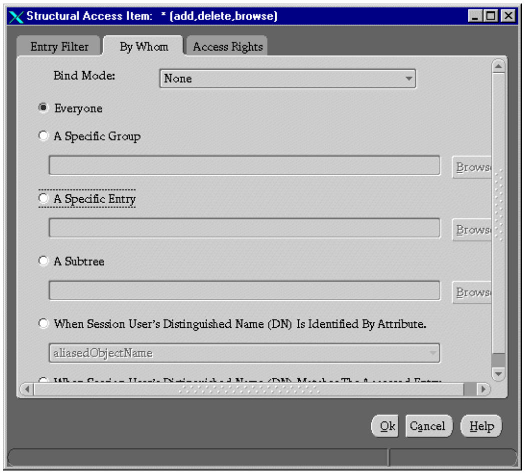
a. In the menu at the left end of the bar, select an attribute.

b. In the menu in the middle of the bar, select one of the following filter options:

Filter	Description
Begins With	To search by using only the first few characters of the attribute's value
Ends With	To search for an entry by using only the last few characters of the specified attribute's value
Contains	To search for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	To search for an entry whose specified attribute is the same as the value you enter
Greater or Equal	To search for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter
Less or Equal	To search for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter
Present	To determine 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 that 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 selected.

4. Select the By Whom tab:



Structural Access Item: * (add,delete,browse)

Bind Mode: None

● Everyone

○ A Specific Group

○ A Specific Entry

○ A Subtree

○ When Session User's Distinguished Name (DN) Is Identified By Attribute.

aliasedObjectName

5. 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:

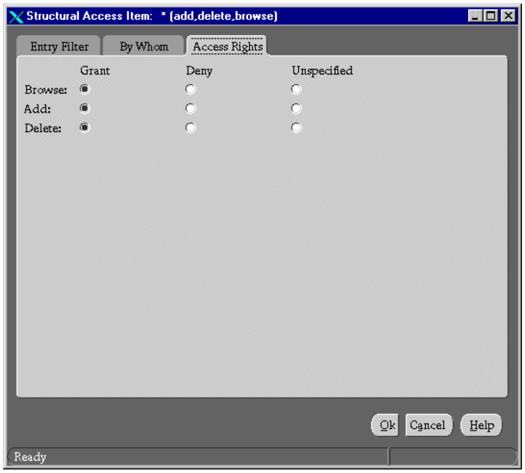
Bind Mode	Description
None	No authentication
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption / decryption is used.
SSL One Way	Only the directory server authenticates itself to the client. 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. Both the client and server send 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.

In the By Whom tab, specify the entity or entities to whom you are granting access:

Entity	Description
Everyone (*)	All who try to access the entry
A specific group (dn)	A previously defined group name
A specific entry (dn)	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

6. Select the Access Rights tab page:



Structural Access Item: * (add,delete,browse)

	Grant	Deny	Unspecified
Browse:	●	○	○
Add:	●	○	○
Delete:	●	○	○

Ready

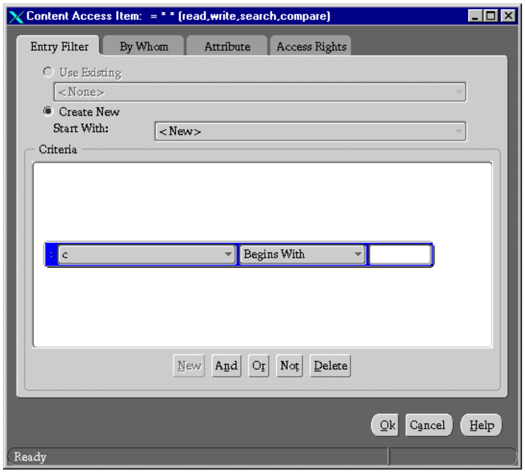
7. In the Access Rights tab, determine what kinds of rights are granted: Browse, Add, or Delete.
8. Click OK.
9. In the main Oracle Directory Manager dialog box, click Apply.

Note: You must click Apply in the main Oracle Directory Manager dialog box in order 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.

Modifying Content Access Items of an Existing ACP

To modify a content access item of an existing ACP:

1. In the navigator pane, expand Subtree Access Management. All of the defined Access Control Policy Points (ACPs) appear in a list below Subtree Access Management in the navigator pane. They also appear in the right pane.
2. Select an ACP under Subtree Access Management in the navigator pane 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.
3. In the Content Access Items window of the Subtree Access Management pane, select the Content Access Item you want to modify.
4. Just below the Content Access Item window, click the Edit button. The Content Access Items dialog box appears:



Content Access Item: = * * (read,write,search,compare)

○ Use Existing: <None>

● Create New

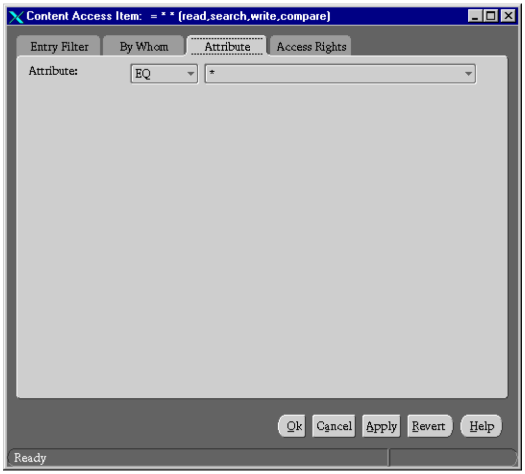
Start With: <New>

Criteria: c | Begins With |

Each tab page contains items you can modify.

5. Specify the items in the Entry Filter tab (if applicable) as described in the section "Modifying Structural Access Items of an Existing ACP" on page 9-28.

6. Select the By Whom tab and specify the items as described in the section "Modifying Structural Access Items of an Existing ACP" on page 9-28.
7. Select the Attribute tab:



Content Access Item: = * * (read,search,write,compare)

Attribute: EQ *

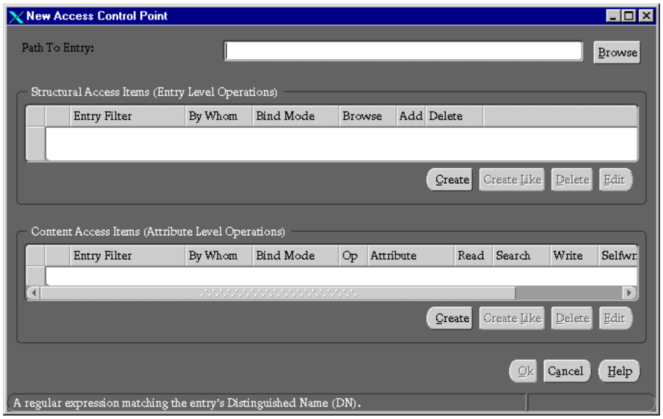
8. From the right menu in the Attribute Tab Page, select the attribute to which you want to grant or deny access.
9. In the left menu in the Attribute Tab Page, select the matching operation to be performed against the attribute. Choices are EQ (Equal (=)) and NEQ (Not Equal (!=)).
10. Select the Access Rights tab and specify the items as described in the section "Modifying Structural Access Items of an Existing ACP" on page 9-28.
11. Click OK.
12. In the main Oracle Directory Manager dialog box, click Apply.

Note: You must click Apply in the main Oracle Directory Manager dialog box in order 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.

Adding an ACP and Creating Access Items

To create a new ACP:

1. In the navigator pane, select Subtree Access Management.
2. On the toolbar, click the Create button. A New Access Control Point window appears:



A regular expression matching the entry's Distinguished Name (DN).

3. In the Path to Entry field, enter the distinguished name (DN) of the entry that will be the ACP.

> **Note:** You can find the DN by looking in the navigator pane for the entry or by clicking the Browse button.

4. To define structural access items (entries), click Create under the Structural Access Items window. The Structural Access Item dialog box appears. Use the tab pages in this dialog box as explained in the section "Modifying Structural Access Items of an Existing ACP" on page 9-28.
5. To define content access items (attributes), click Create under the Content Access Items field. The Content Access Item dialog box appears. Use the tab pages in this dialog box as explained in the section "Modifying Content Access Items of an Existing ACP" on page 9-32.
6. Click OK to close this dialog box and return to the main Oracle Directory Manager dialog box.

Note: You must click Apply in the main Oracle Directory Manager dialog box in order 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.

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.

Managing ACPs: An Example

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 each user

Create a New ACP

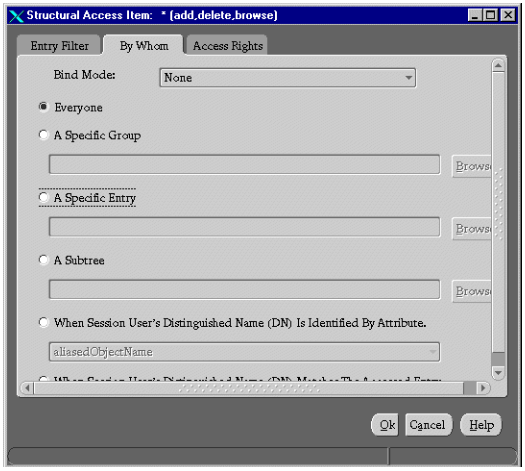
1. In the navigator pane, select Subtree Access Management. A list of ACPs appears in the right pane.
2. Click Create at the bottom of the right pane. A New Access Control Point dialog box appears.
3. In the PathTo Entry field, enter the DN where you want the ACP. The ACIs within the ACP will apply to all entries below and including that DN.

Structural Access Items To set the access rights for an entry, click Create under the Structural Access Items field.

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.

1. Select the By Whom tab to define the subject of the ACI.

   The By Whom page appears:



Structural Access Item: * (add,delete,browse)

Bind Mode: None

● Everyone

○ A Specific Group

○ A Specific Entry

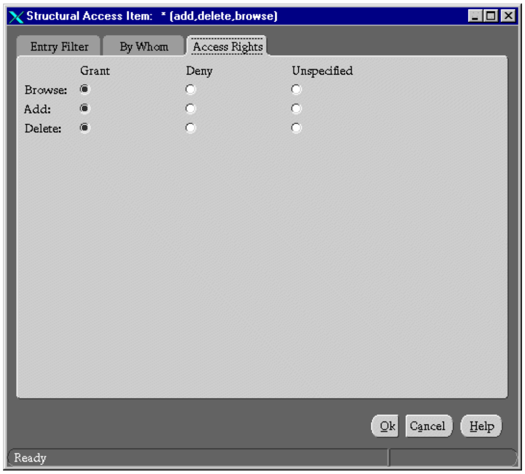
○ A Subtree

○ When Session User's Distinguished Name (DN) Is Identified By Attribute.

aliasedObjectName

2. Use this page to define the subject of the ACI. In the Bind Mode field, select the authentication mode appropriate to your environment. To create access rights for everyone, click the Everyone button. Click OK to set these permissions.

3. Select the Access Rights tab page. By default, all rights—browse, add, and delete—are granted. Change the access rights so that Everyone can browse all entries, but cannot add or delete them. The page will look like the following:



Structural Access Item: * (add,delete,browse)

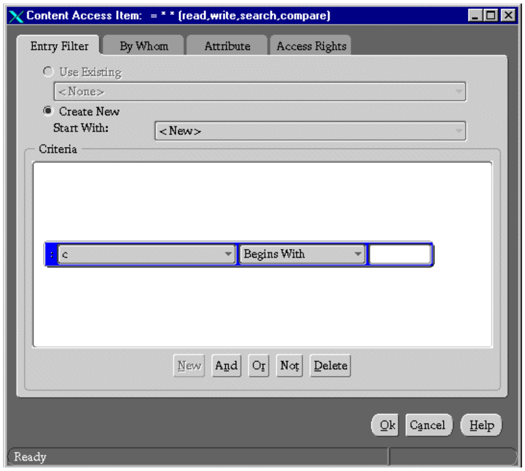
	Grant	Deny	Unspecified
Browse:	●	○	○
Add:	●	○	○
Delete:	●	○	○

4. Click OK to set the these permissions and close the dialog box.

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.

1. To define the content access items, click Create below the Content Access Items field.

The Content Access Items dialog box appears:



Content Access Item: = * * (read,write,search,compare)

○ Use Existing

<None>

● Create New

Start With: <New>

Criteria

c

Begins With

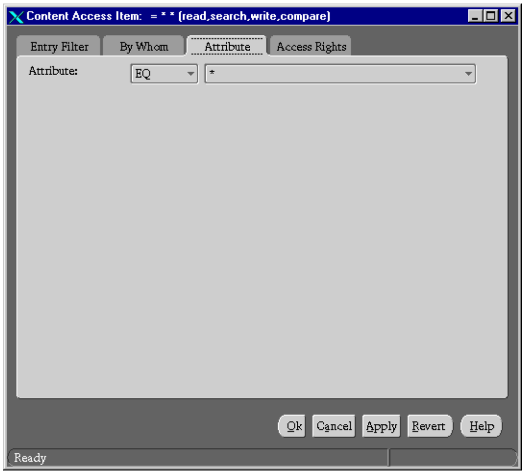
Ready

It is similar to the Structural Access Items dialog box, but it has four tabs: Entry Filter, By Whom, Attribute, Access Rights.

2. Because you want this ACI to apply to all entries under the ACP, do not use the Entry Filter tab. Select the By Whom tab.

3. On the By Whom tab page, click the Everyone button, then click OK.

4. Select the Attribute tab page:



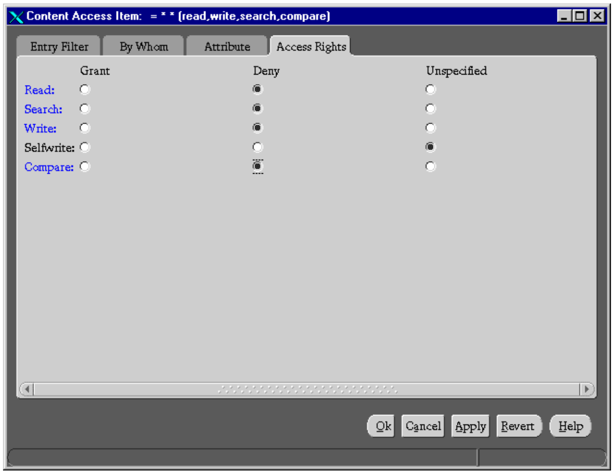
Content Access Item: = * * (read,search,write,compare)

Attribute: EQ *

This page has two fields.The first field has two choices: EQ (equals) and NEQ (not equals). The second field sets the attribute; select it from the menu.

5. Select EQ and select userPassword.

6. Select the Access Rights tab page. By default, all permissions are granted. Change the permissions so that Read, Search, and Write and Compare are denied. The result looks like the following:



Content Access Item: = * * (read,write,search,compare)

	Grant	Deny	Unspecified
Read:	○	●	○
Search:	○	●	○
Write:	○	●	○
Selfwrite:	○	○	●
Compare:	○	●	○

7. Click OK to set these permissions and close the dialog box.

   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 field, click Create. This displays the Content Access Items dialog box.
2. Select the By Whom tab.
3. At the bottom of the By Whom tab page, click the button labeled When Session User’s Distinguished Name (DN) Matches the Accessed Entry, then click OK.
4. Select the Attribute tab. The Attribute page appears.

   This tab page has two fields.The first has two choices: EQ (equals) and NEQ (not equals). The second sets the attribute; select it from the menu.
5. On this page, select EQ and userPassword.
6. Select the Access Rights tab.

7. On the Access Rights tab page, grant access to read, search, write, and compare. Leave selfwrite unspecified.
8. 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, and write it, as well as compare it.

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. Press Create under the Content Access Items field to display the Content Access Items.
2. Select the By Whom tab.
3. On the By Whom tab page, select Everyone. Click OK.
4. Select the Attribute tab page. This page has two fields.The first field has two choices: EQ (equals) and NEQ (not equals). The second field sets the attribute; select it from the menu.
5. On 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.
6. Select the Access Rights tab.
7. On the Access Rights tab page, grant access to read, search, and compare. Deny write access. Leave selfwrite unspecified.
8. 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. Click Create under the Content Access Items field to display the Content Access Items.
2. Select the By Whom tab.

3. At the bottom of the By Whom tab page, select When Session User's Distinguished Name (DN) Matches the Accessed Entry. Click OK.
4. Select the Attribute tab. This tab page has two fields.The first field has two choices: EQ (equals) and NEQ (not equals). The second field sets the attribute.
5. On this 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.
6. Press the Access Rights tab.
7. On the Access Rights tab page, grant access to read, search, and compare. Deny write access. Leave Selfwrite unspecified.
8. Click OK.

Consider other access restrictions you might want to implement. Your directory might contain many entries and attributes that should not be available to everyone.

Note: You must click Apply in the main Oracle Directory Manager screen in order 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.

Granting Entry-Level Access

To grant local access, or access to one entry, you can use either Oracle Directory Manager or command line tools. This section explains how to grant access using Oracle Directory Manager.

To grant entry-level access by using Oracle Directory Manager:

1. In the navigator pane, select Entry Management, then click the entry to display it in the right pane. Alternatively, in the right pane, use the search panel to display the entry that you intend to change.
2. You can change the structural access and content access items for the entry in the right pane by selecting the Local Access tab.
3. Create and edit local ACIs in the Structural Access Item and Content Access Item fields.
4. Once you have made the changes, click Apply in the main Oracle Directory Manager dialog box to save your changes.

> **Note:** You must click Apply in the main Oracle Directory Manager screen in order 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 the beginning of this chapter, directory access control policy information is represented as user modifiable operational attributes. Hence, one can manage directory access control by setting and altering values of these attributes using the ldapmodify command. Any tool, including the command line tools that support the ldapmodify operation, can be used for the purpose.

In order 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:

- "Using LDAP Data Interchange Format (LDIF)" on page A-2 for information about how to format input using **LDAP Data Interchange Format (LDIF)**, the required input format for line mode commands
- "ldapmodify" on page A-11 for information about how to run ldapmodify
- Appendix E for information about the format or syntax of ACI

Managing Access Control: Examples

This section provides several examples using command line tools. The examples are covered in the following sections:

- Setting Up an Inheritable ACP by Using ldapmodify
- Setting Up Entry-Level ACIs by Using ldapmodify
- Typical Access Control Policies

Setting Up an Inheritable ACP by Using ldapmodify

This example sets up subtree access permissions in an `orcACI` at the **Root DSE**. Because this example refers to the `orclACI` attribute, this access directive governs all the entries in the DIT that are located below the entry in which this access directive resides. Specifically, this example replaces the values in `orclACI`.

Note the mandatory presence of the << EOF characters.

```
ldapmodify -v -h $1 -D "cn=Directory Manager, o=IMC, c=US" -w "controller" <<
EOF
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)

EOF
```

Setting Up Entry-Level ACIs by Using ldapmodify

The example in this section sets up entry-level access permissions in the `orclEntryLevelACI` attribute. Because this resides in the `orclEntryLevelACI` attribute, this ACL governs only the entry in which it resides. Note the mandatory presence of the << EOF characters.

```
ldapmodify -v -h $1 -D "cn=Directory Manager, o=IMC, c=US" -w "controller" <<
EOF
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)

EOF
```

Typical Access Control Policies

This section shows some advanced and typical examples of access control policies.

Example 9–1 Using Wild Cards

The following example is an illustration of the use of wild cards (*) in the *object* and *subject* parts:

`orclACI` attribute in the ACP at `dc=com`

```
access to entry by * (browse)
access to attr=(*) by * (search, read)
```

This access directive grants read access to Everyone. Note that, in order to allow reading the attributes, browse permissions are granted on the entries.

Example 9–2 Selecting Entries by DN

The following example shows the use of a regular expression to select the entries by DN in two access directives.

Read access is granted to entries under the `dc=oracle,dc=com` subtree, except for those entries under the `dc=us,dc=oracle,dc=com`, to which search access is granted.

`orclACI` attribute in the ACP at `dc=oracle,dc=com`:

```
access to entry by * (browse)
access to attr=(*) by * (search, read)
```

`orclACI` attribute in the ACP at `dc=us,dc=oracle,dc=com`:

```
access to entry by * (browse)
access to attr=(*) by * (none)
```

Example 9–3 Using Attribute and Subject Selectors

The next 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=oracle,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 subtree can browse, but those outside have no access to the subtree.
- The salary attribute can be modified by one's manager, viewed by oneself, but has no access to anyone else.
- The `userPassword` attribute can be viewed and modified by oneself and the administrator. Others can compare.
- The `homePhone` attribute can be read and written by oneself and viewed by others.
- For all other attributes, the administrator can compare, search on, read and modify. Others cannot update.

`orclACI` attribute in the ACP at `dc=us,dc=oracle,dc=com`:

```
access to entry
      by dn="cn=admin, dc=us,dc=oracle,dc=com" (browse, add, delete)
      by dn=".*, dc=us,dc=oracle,dc=com" (browse)
       by * (none)

 access to attr=(salary)
       by dnattr=(manager) (search, read, write, compare)
       by self (search, read, compare)
        by * (none)

 access to attr=(userPassword)
       by self (search, read, write, compare)
       by dn="cn=admin, dc=us,dc=oracle,dc=com" (search,read, write, compare)
        by * (compare)
```

```
access to attr=(homePhone)
      by self (search,read, write, compare)
       by * (read)

access to attr != (salary, userPassword, homePhone)
     by dn="cn=admin, dc=us,dc=oracle,dc=com" (compare, search, read, write)
      by * (compare, search, read)
```

Example 9–4 Using Selfwrite

Sometimes it is useful to permit a particular DN to add or remove itself from an attribute. For example, if you would like to create a group and allow people to add and remove only their own DN from the member attribute, you could accomplish it with an access directive like this:

```
access to attr=(member) by dnattr=(member) (selfwrite)
```

The `dnattr=(`*`subject)`* selector says that the access applies to entities listed in the member attribute. The `(selfwrite)` access selector says that such members can only add or delete their own DN from the attribute, not other values.

10

Managing Directory Replication

Replication is the mechanism that maintains exact duplicates of specified naming contexts on multiple nodes.

Note: For Release 2.0.6, you can use Oracle Internet Directory replication only if you have installed Advanced Symmetric Replication, which ships with Oracle 8*i* Enterprise Edition. Advanced Symmetric Replication does not ship with Oracle 8*i* Standard Edition.

This chapter addresses:

- Installing and Configuring Replication
- Adding a Replication Node
- Conflict Resolution
- The Replication Process

See Also: "Distributed Directories" on page 2-10 to review the conceptual material before setting up replication

Installing and Configuring Replication

This section describes how to install and initialize Oracle Internet Directory replication server software on a node.

Each node in a group of **DSA**s holds an updatable copy, also called an updatable replica, of the same set of directory 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 10-17.

To install and configure a replication group, follow the general steps listed below.

- Step 1: Install Oracle Internet Directory on All Nodes in the DRG
- Step 2: Configure Database Parameters for ASR on All Nodes
- Step 3: Decide Which Node Will Serve as the ASR Master Definition Site (MDS)
- Step 4: At the MDS, Set Up ASR for a Directory Replication Group
- Step 5: Start Oracle Directory Server Instances on All the Nodes
- Step 6: Configure Replication
- Step 7: Start the Replication Servers on All the Nodes

Note: In this release of Oracle Internet Directory, procedures and tools to support the following deployment scenarios are *not* available:

- Removing an Oracle Internet Directory server from an existing DRG
- Creating an environment (directory network) consisting of more than one DRG

Step 1: Install Oracle Internet Directory on All Nodes in the DRG

Refer to Oracle Internet Directory *Installation Guide*. Note that the typical installation of the Oracle 8*i Enterprise* Edition, which is required for the Oracle Internet Directory, includes Oracle **Advanced Symmetric Replication (ASR)**. By contrast, a typical installation of Oracle 8*i Standard* Edition does *not* include ASR.

Step 2: Configure Database Parameters for ASR on All Nodes

If you are going to run replication, you must change the values for three parameters by following these steps:

1. Stop the database server
2. Change the following parameters:
   - JOB_QUEUE_PROCESSES = 4
   - GLOBAL_NAMES = TRUE
   - OPEN_LINKS = 4
3. Restart the database for the changes to take effect.

See: *Oracle8i Administrator's Guide*

Step 3: Decide Which Node Will Serve as the ASR Master Definition Site (MDS)

A **Master Definition Site (MDS)** is any of the Oracle Internet Directory databases from which the administrator is going to run the configuration scripts. You should be able to connect to the MDS database and all other nodes that constitute the DRG using Net8.

See Also: "Prepare the Net8 Environment for Replication" on page 10-4

Step 4: At the MDS, Set Up ASR for a Directory Replication Group

The following sections lead you through installing and configuring ASR through Oracle Internet Directory installation scripts. Advanced ASR users may prefer to configure ASR through the Oracle8*i* Replication Manager Tool.

See Also: Oracle8*i* Server replication documentation and the online help for Oracle8*i* Replication Manager Tool for information on configuring ASR with the Oracle8*i* Replication Manager Tool

Setting up the Oracle Advanced Symmetric Replication (ASR) environment to establish a Directory Replication Group (DRG) requires you to perform the two tasks described in the following sections:

- "Prepare the Net8 Environment for Replication" on page 10-4
- "Configure Oracle ASR For Directory Replication" on page 10-6

Prepare the Net8 Environment for Replication

See Also: *Net8 Administrator's Guide.*

Execute the following steps on *all nodes* in the Directory Replication Group.

1. Configure `sqlnet.ora`.

   The `sqlnet.ora` file in `$ORACLE_HOME/network/admin` should contain the following parameters at minimum:

   ```
   names.directory_path = (TNSNAMES)
   names.default_domain = domain
   ```

2. Configure `tnsnames.ora`.

   The `tnsnames.ora` file in `$ORACLE_HOME/network/admin` 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))
   ```

   Note: You may domain-qualify the net service name (for example, sales.com). If you domain-qualify it, be sure that the domain matches the domain specified by the NAMES.DEFAULT_DOMAIN parameter in the `sqlnet.ora` file.

3. For long transactions, 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.

   See Also: *Oracle8i Administrator's Guide.* for instructions on increasing size of the table spaces and segments

   a. Create table space 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.

      Execute SQL*Plus by typing the following command,

      ```
      sqlplus system/system_password@net_service_name
      ```

      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 commands above for each rollback segment entered in `initsid.ora.`

4. Modify the parameters in file `initsid.ora`.

   Type the following lines in the `initsid.ora` file:

   ```
   rollback_segments = (rollback_segment_name_1, rollback_segment_name_2 ...)
   JOB_QUEUE_PROCESSES = number number of LDAP nodes - 1, at a minimum
   SHARED_POOL_SIZE = 20000000
   ```

   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 system 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.
>
> For more information on SGA, see *Oracle8i Concepts* and *Oracle8i Administrator's Guide*.

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 a command prompt:

   ```
   lsnrctl> set password password
   lsnrctl> 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:

   ```
   lsnrctl> 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.

Configure Oracle ASR For Directory Replication

To configure ASR at the MDS and all nodes in the replication group, complete the following steps *from the MDS*:

1. Log on to the Oracle Internet Directory software owner account at the UNIX level.
2. Change the directory to `$ORACLE_HOME/ldap/bin`.

Note: Before proceeding to the next step, connect as system user to all nodes, including MDS, from MDS to ensure the following:

- Oracle Internet Directory database is up and running
- Oracle Internet Directory listener is up and running
- Connect string is correct
- System password is correct

See Also:

- *Oracle8i Administrator's Guide* for instructions on ensuring that the database and listener are running
- *Net8 Administrator's Guide* for instructions on ensuring that the connect string is correct

3. Run the following script from MDS:

```
ldaprepl.sh -asrsetup
```

This script executes a number of operations.

- It configures the Master Definition Site.
- It configures the remote master sites. A remote master site is any site other than the Master Definition Site that participates in ASR replication.
- It configures replication "Push" jobs at all sites.
- It resumes replication at the Master Definition Site.
- It verifies that all steps have completed successfully.

As the script runs, it asks for the information in Table 10–1, first for the Master Definition Site then for the master sites.

Table 10–1 ASR Setup Information

Information	Definition
Host Name of MDS	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, it asks if there is another master site. Enter Y or N. When you enter N, to indicate that you have identified all sites, it shows you a table of the information you have provided, and asks for confirmation. If it is not correct, press N. The script will start again at the beginning, asking about the Master Definition Site 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, the script begins configuring the sites.

This process may 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, you will have to start at the beginning. Interrupting the process will not negatively affect your re-installation.

Troubleshooting Tip: If the configuration process fails, do the following:

- Check the `$ORACLE_HOME/ldap/admin/logs/ldaprepl.log` file to see the status.
- In the directory `$ORACLE_HOME/ldap/admin`, check the status of replication jobs by running the following command:

  ```
  sqlplus system/password@database_connect_string@ldaplogq.sql
  ```

Run the above 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 ASR set up failed. In this case, you may:

- Determine a solution from error message information by consulting an expert in Advanced Symmetric Replication (ASR)
- Consult the troubleshooting chapter in *Oracle8i Replication*,
- Run the script from the beginning

Note: If you have large initial data requirements, 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. For bulkload syntax and usage notes, see "bulkload" on page A-20.

Step 5: Start Oracle Directory Server Instances on All the Nodes

Run the following command:

```
oidctl connect=net_service_name_of_new_node server=oidldapd instance=instance_
number_of_ldap_server flags="-p port" start
```

See Also: Chapter 5 for more information on starting an Oracle Directory Server **instance**

Step 6: Configure Replication

You need to configure two different kinds of entries for replication, and these are described in the following sections:

- "Replication Server Configuration Parameters" on page 10-10
- "Replication Agreement Parameters" on page 10-13

Replication agreements are entries that list the member nodes within a replication group that share their changes. Replication agreements are referenced by Oracle Directory Replication Server configuration parameters that load when the replication server runs.

Because the Oracle 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—that is, you can alter the contents of the configuration entries and agreement entries through the command line tools, such as ldapadd and ldapmodify, or you can view and modify the agreements by using Oracle Directory Manager. This section explains both approaches.

Important: When you install and configure replication the first time, you must inform the Oracle 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. This is explained in "Replication Agreement Parameters" on page 10-13.

Replication Server Configuration Parameters

The Oracle 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 which control replication processing. You can modify some of these attributes. Note that the last parameter in the list specifies a replication agreement. In this release, only one replication agreement is possible.

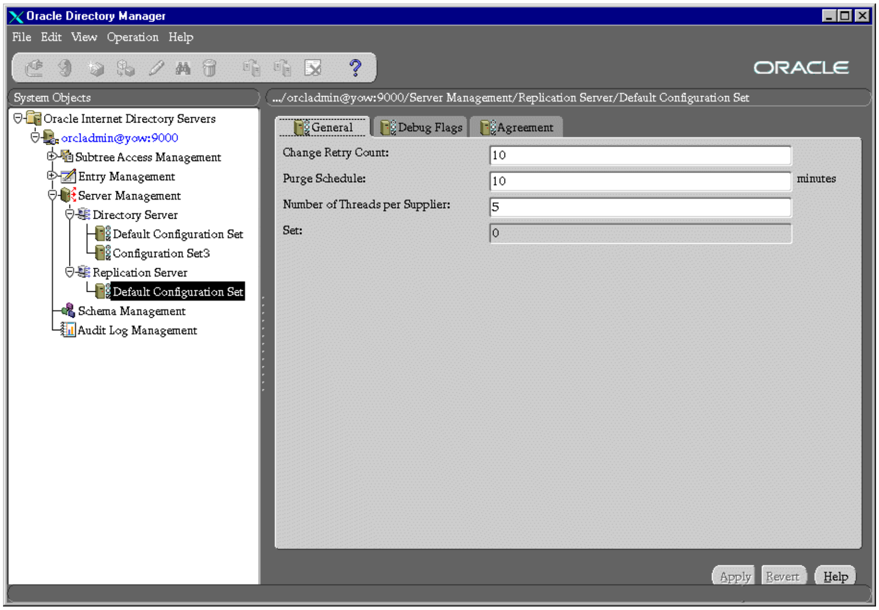
Table 10–2 lists and describes the replication server configuration parameters.

Table 10–2 Replication Server Configuration Parameters

Parameter name	Description	Default Values	Modifiable?
orclChangeRetryCount	Single-valued attribute. The number of processing retry attempts for a change-entry before being dropped. You cannot use a negative value for this parameter.	default: 10	Yes
orclPurgeSchedule	Single-valued attribute. 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. Specifies purge (garbage collection) interval in minutes. You cannot use a negative value for this parameter.	default: 10 minutes	Yes
orclThreadsPerSupplier	Number of worker threads the Oracle Directory Replication Server provides for each supplier for change log processing. You cannot use a negative value for this parameter.	5	Yes
orclDirReplGroupAgreement	Multi-valued attribute. Identifies the symmetrical replication agreements for which this server is responsible.	orclagreementid=000001,cn=orclreplagreements	No

Viewing and Modifying Replication Configuration Parameters by Using Oracle Directory Manager

1. In the navigator pane expand 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:



.../orcladmin@yow:9000/Server Management/Replication Server/Default Configuration Set

Change Retry Count: 10

Purge Schedule: 10 minutes

Number of Threads per Supplier: 5

Set: 0

Configuration parameters appear in the General and Debug Flags tab pages. You can use these tab pages to view replication configuration parameters, and modify many of them. The following tables describe the fields in each tab page.

Table 10–3 General Tab Page

Field	Description
Change Retry Count	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	The number of minutes in between running garbage collection. The replication garbage collection thread removes entries that are already applied or have been dropped as candidate changes. The default is 10.

Table 10–3 General Tab Page

Field	Description
Orcl Threads Per Supplier	Number of worker threads the Oracle Directory Replication Server provides for each supplier for change log processing. The default is 5.
Set	The configuration identifier

Table 10–4 Debug Flags Tab Page

Field	Description
Debug Level	The level of debugging that is set for this replication server. This number will be 32768 for replication-specific tracing.
Various debug flags	The only values you should change are the Trace Function Calls or the Any check box. Click these flags if you want to see more data during debugging.

Modifying Replication Configuration Parameters by Using Command Line Tools To modify replication configuration parameters by using command line tools, use the commands explained in "ldapmodify" on page A-11.

Modifying the Garbage Collection Interval To modify the interval for garbage collection in replication, run ldapmodify, referencing an LDIF-formatted file. Before running this command, prepare an input file using LDIF format.

The LDAP command to apply the input file is as follows:

```
ldapmodify -h host -p port -f filename
```

Example: A typical input file (using LDIF format) to modify the garbage collection interval parameter consists of the following lines:

```
dn: cn=configset0, cn=osdrepld, cn=subconfigsubentry
changetype:modify
replace:orclPurgeSchedule
orclPurgeSchedule:30
```

This procedure changes the garbage collection interval from the default of 10 minutes to 30 minutes.

Example: A typical input file (using LDIF format) to modify the retry counts parameter consists of the following lines:

```
dn: cn=configset0, cn=osdrepld, cn=subconfigsubentry
changetype:modify
replace:orclChangeRetryCount
orclChangeRetryCount:5
```

This procedure changes 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 file.

> **Important Note:** To configure replication, you must modify the attribute orclDirReplGroupDSAs to contain the values of the nodes participating in symmetrical replication. For instructions on how to modify any of these parameters, see "Modifying Replication Agreement Parameters Using Command Line Tools: A Sample" on page 10-15.

Modifying the Number of Worker Threads Used in Change Log Processing To modify the number of worker threads used in change log processing:

1. Edit `mod.ldif` as follows:

   ```
   Dn: cn=configset0, cn=osdrepld, cn=subconfigsubentry
   Changetype: modify
   Replace: orclthreadspersupplier
   orclthreadspersupplier: new_number_of_worker_threads
   ```

2. Use ldapmodify to update the replication server configset0 parameter value as follows:

   ```
   ldapmodify -h host -p port -f mod.ldif
   ```

3. Restart the replication server.

   > **See Also:** "Restarting Directory Server Instances" on page 3-8 for instructions on restarting the replication server

Replication Agreement Parameters

In the parameter DirectoryReplicationGroupDSAs, type all of the host names of the DSAs in the DRG. Make sure this information is identical in all the nodes.

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 using Oracle Directory Manager. For Oracle Internet Directory Release 2.0.6, only one replication agreement can be used.

Table 10–5 lists and describes the replication agreement parameters.

Table 10–5 Replication Agreement Parameters

Field in Oracle Directory Manager	Parameter	Description	Default Values	Modifiable?
Agreements ID	orclAgreementID	Unique identifier for a replication agreement.	000001	No
Excluded Naming Contexts	orclExcludedNamingcontexts	Multi-valued attribute. Specifies naming contexts excluded from this replication agreement.		Yes
Replication Group Nodes	orclDirReplGroupDSAs	Multi-valued attribute. Specifies nodes participating in symmetrical replication agreement. *Nodes that you specify here share updates.*		Yes
Update Schedule	orclUpdateSchedule	Specifies replication update interval in minutes.	1	Yes
Replication Protocol	orclReplicationProtocol	Specifies the replication protocol used in this replication agreement. The supported protocol is ASR.	ODS_ASR_1.0	No

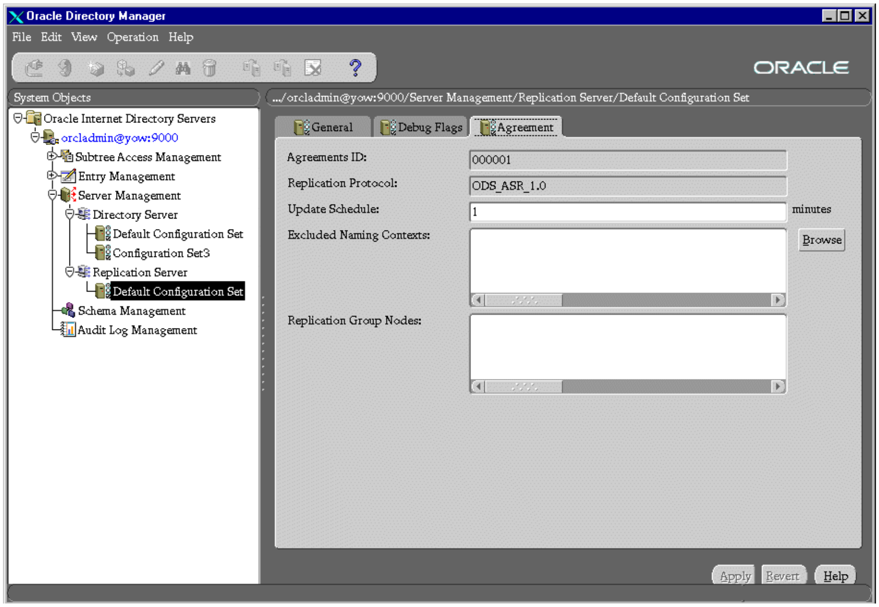
Note: Before you modify replication agreement parameters, be sure that you have started the Oracle Internet Directory on all nodes.

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 panel expand Server Management > Replication Server. Select Default Configuration Set.

2. In the right pane, select the Agreement tab to display the replication agreement:



Agreements ID: 000001

Replication Protocol: ODS_ASR_1.0

Update Schedule: 1 minutes

Excluded Naming Contexts:

Replication Group Nodes:

The fields in this tab page are described in Table 10–5 on page 10-14. You can view the parameters and modify some of them by double-clicking the attributes.

3. When you have finished making your changes, click the Apply button. If you want to go back to using the values that appeared when you first opened this pane, click the Revert button.

Modifying Replication Agreement Parameters Using Command Line Tools: A Sample To add more nodes to the values in a replication agreement entry, run ldapmodify at the command line, referencing an LDIF-formatted file. Before running this command, prepare an input file using LDIF format.

The LDAP command to apply the input file is as follows:

```
ldapmodify -h host -p port -f filename
```

A typical input file (using LDIF format) to add two more nodes to a replication agreement consists of the following lines:

```
dn: orclagreementid=000001, cn=orclreplagreements
changetype:modify
add:orcldirreplgroupdsas
orcldirreplgroupdsas:hollis
orcldirreplgroupdsas:eastsun-11
```

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`.

Because this release of the Oracle Directory Replication Server supports only one configuration set, you do not need to specify a configuration set.

Step 7: Start the Replication Servers on All the Nodes

Type the following command:

```
oidctl connect=db_connection_string server=oidrepld instance=1
   flags="-h host -p port" start
```

> **See Also:** Chapter 5 for information on starting the replication servers

Toggling the Change-Log Flag

You can turn off change-logging, which occurs in the Oracle Internet Directory server, by toggling the default value of the `-l` flag in the line-mode run 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.

Toggling the Multi-Master Flag

You can turn off the multi-master flag, which occurs in the replication server, by toggling the default value of the *-m* flag in the line-mode run 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 multi-master option controls conflict resolution, which serves no purpose if you are deploying a single master.

Adding a Replication Node

There are two ways to add a new node to a replication network. The easier of the two procedures is described in this section. Use this procedure unless your directory is very large. If your directory has more than a million entries, use the method described in Appendix B.

> **Note:** Use the procedure in this section to add a new node to an existing replication group. To create a new replication group, follow the instructions in "Installing and Configuring Replication" on page 10-2.

> **Note:** Before you add a replication node, prepare the Net8 environment. For instructions, see "Prepare the Net8 Environment for Replication" on page 10-4.

To add a replication node to a directory with less than a million entries, follow these steps, each of which is more fully described in the next few pages.

- Step 1: Stop the Replication Server on All Nodes
- Step 2: Configure the New Node into the LDAP Replication Group on All the Existing Nodes
- Step 3: Identify a Sponsor Node and Switch the Sponsor Node to Read-Only Mode
- Step 4: Back Up the Sponsor Node by Using ldifwrite
- Step 5: Perform ASR Add Node Setup
- Step 6: Switch the Sponsor Node to Updatable Mode
- Step 7: Start the Replication Server on All Nodes Except the New Node
- Step 8: Load Data into the New Node by Using bulkload
- Step 9: Start LDAP Server on the New Node
- Step 10: Configure the LDAP Replication Agreement on the New Node

- Step 11: Start the Replication Server on the New Node

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`

When typing a command, be sure that `$ORACLEHOME/`*`variable`* is in the path.

Step 1: Stop the Replication Server on All Nodes

Run the following command against each node in the LDAP replication group:

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

Step 2: Configure the New Node into the LDAP Replication Group on All the Existing Nodes

Create an LDIF file, such as `add_node.ldif`, as in the following example:

```
dn: orclagreementid=000001, cn=orclreplagreements
changetype: modify
replace: orcldirreplgroupdsas
orcldirreplgroupdsas: host_name_of_the_new_node
orcldirreplgroupdsas: host_name_of_old_node1
orcldirreplgroupdsas: host_name_of_old_node2
.
.
.
orcldirreplgroupdsas: host_name_of_old_noden
```

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
```

Step 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.

Edit `change_mode.ldif` to the following:

```
dn:
changetype:modify
replace:orclservermode
orclservermode:r
```

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= db_connection_string server=oidldapd restart
```

This restarts all running LDAP severs on the sponsor 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 Master Definition Site may be the same node.

Step 4: Back Up the Sponsor Node by Using ldifwrite

Because this may take a long time, you may start "Step 5: Perform ASR Add Node Setup" while backup is in process.

You can backup the sponsor node in one of two ways:

- Using ldifwrite

  This method supports filter-based backup, and the process can be fully automated. The generated file can be used for partial replication. However, backup may take up to seven hours for a directory with one million entries. Enter the following command:

  ```
  ldifwrite -c db_connect_string -b "" -f output_ldif_file
  ```

- Using cold backup

  This method, described in Appendix B, cannot be fully automated, and cannot be reused for partial replication. However, cold backup takes much less time for a directory with one million entries.

Step 5: Perform ASR Add Node Setup

You can perform this step at the same time as you are performing "Step 4: Back Up the Sponsor Node by Using ldifwrite".

From the MDS, run the following script:

```
ldaprepl.sh -addnode
```

This script executes a number of operations.

- It quiesces ASR at the MDS and other existing master sites.
- It configures the master sites and the new node. A master site is any site other than the Master Definition Site 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 10–6, first for the Master Definition Site then for the existing master sites.

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, press N. The script then starts again at the beginning, asking the same information. If the information is correct and you enter Y, the script begins configuring the sites.

Table 10–6 lists and describes the information for which the script prompts you.

Table 10–6 ASR 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`
system password	system password

This process may take a long time, depending on your system resources and the size of your DRG. The script will keep you informed of its progress.

Note: If for any reason you must interrupt the process before it is complete, you will have to start again at the beginning.

Troubleshooting Tip: If the process fails, do the following:

- Check the `$ORACLE_HOME/ldap/admin/logs/ldaprepl.log` file to see the status.
- Move to the directory `$ORACLE_HOME/ldap/admin` and check the status of replication jobs by running the following command:

  ```
  sqlplus system/password@database_connect_string@ldaplogq.sql
  ```

Run the above 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 ASR set up failed. In this case, you may:

- Determine a solution from error message information by consulting an expert in Advanced Symmetric Replication (ASR)
- Consult the troubleshooting chapter in *Oracle8i Replication*,
- Run the script from the beginning

Step 6: Switch the Sponsor Node to Updatable Mode

Edit `change_mode.ldif` to the following:

```
dn:
changetype:modify
replace:orclservermode
orclservermode:rw
```

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=db_connection_string server=oidldapd restart
```

> **Note:** Step 6 is very similar to Step 3. The only difference is that the `orclservermode` in `change_mode.ldif` is `rw` instead of just `r`.

Step 7: Start the Replication Server on All Nodes Except the New Node

Do this by entering the following command:

```
oidctl connect=db_connection_string server=oidrepld instance=1
flags="-h host -p port" start
```

Verify that no processes are running on the new node.

Step 8: Load Data into the New Node by Using bulkload

Do this by entering the following command:

```
bulkload.sh -connect db_connect_string_of_new_node -generate -load
-restore absolute_path_to_the_ldif_file_generated_by_ldifwrite
```

Step 9: Start LDAP Server on the New Node

Do this by entering the following command:

```
oidctl connect=db_connect_string_of_new_node server=oidldapd
instance=1 flags="-p port" start
```

Step 10: Configure the LDAP Replication Agreement on the New Node

Create an LDIF file, such as `add_node.ldif`, as in the following example:

```
dn: orclagreementid=000001, cn=orclreplagreements
changetype: modify
add: orcldirreplgroupdsas
orcldirreplgroupdsas: host_name_of_the_new_node
orcldirreplgroupdsas: host_name_of_old_node1
orcldirreplgroupdsas: host_name_of_old_node2
.
.
.
orcldirreplgroupdsas: host_name_of_old_noden
```

Run the following command against the new node:

```
ldapmodify -h host_name_of_the_new_node -p port -f add_node.ldif
```

Step 11: Start the Replication Server on the New Node

Do this by running 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
```

Conflict Resolution

Multi-master replication enables updates to multiple directory servers. Conflicts are detected whenever the replication server attempts to apply remote changes from a supplier to a consumer that holds conflicting data.

Four kinds of LDAP operations can lead to conflicts:

- Addition
- Deletion
- Modification
- modifyrdn / modifydn

These kinds of operations can be grouped into two categories described in the following sections:

- Entry-Level Conflicts
- Attribute-Level Conflicts

Entry-Level Conflicts

Entry-level conflicts are caused when the replication server attempts to apply a change to the consumer directory. Such a change could be one of the following:

- Adding an entry that already exists
- Deleting an entry that does not exist
- Modifying an entry that does not exist
- Applying a “modifyrdn” operation, but the DN does not exist

These conflicts can be difficult to resolve because the reasons for their existence may be complex. For instance, if an entry does not exist, and this causes a replication conflict to log, that may mean:

- The entry has been moved to a different location.
- The addition of that entry has not yet arrived.
- The entry has been deleted.
- The entry never existed.

If an entry exists and it should not, that may mean:

- The entry was added earlier.
- The entry was moved here by a modifydn operation.

Attribute-Level Conflicts

Attribute-level conflicts are caused when two directories are updating the same attribute with different values at different times. If the attribute is single-valued, 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 Oracle Directory Server replication application attempts to resolve all conflicts that it encounters, utilizing the following logic:

1. The conflict is detected when a change is applied, and an error is encountered.
2. The replication process attempts to reapply the change a configurable number of times or repetitively for a configurable amount of time after a configurable waiting period.
3. The replication process reaches the retry limit without successfully applying the change. It then flags the change as a conflict and moves the change to a low-priority, human intervention queue. Changes are applied according to the time unit you specified in the `orclupdateschedule` parameter in the replication agreement multiplied by 100. Before it moves the change, the 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 large multi-valued variables would have a significant negative impact on performance. System Administrators must be careful to avoid updating from more than one master at a time.

Manual Resolution of Conflicts

If a conflict has been written into the log file, the system is not able to resolve it following its resolution procedure. To avoid further replication change conflicts arising from earlier unapplied changes, it is important to monitor the log files regularly.

To monitor replication change conflicts, examine the contents of the replication log file. New conflict records are written to the file. You can distinguish between messages by the timestamp.

Sample Conflict Resolution Messages

Conflict resolution messages, samples of which are shown below, are logged in file oidrepld00.log. The result of each attempt to resolve the replication conflict is displayed at the end of each conflict resolution message.

```
1999/08/03::10:59:05:  ************ Conflict Resolution Message ************
1999/08/03::10:59:05:  Conflict reason: Attempted to modify a non-existent
entry.
1999/08/03::10:59:05:  Change number:1306.
1999/08/03::10:59:05:  Supplier:eastlab-sun.
1999/08/03::10:59:05:  Change type:Modify.
1999/08/03::10:59:05:  Target
DN:cn=ccc,ou=Recruiting,ou=HR,ou=Americas,o=IMC,c=US.
1999/08/03::10:59:05:  Result: Change moved to low priority queue after failing
on 10th retry.

1999/08/03::10:59:05:  ************ Conflict Resolution Message ************
1999/08/03::10:59:05:  Conflict reason: Attempted to add an existing entry.
1999/08/03::10:59:05:  Change number:1209.
1999/08/03::10:59:05:  Supplier:eastlab-sun.
1999/08/03::10:59:05:  Change type:Add.
1999/08/03::10:59:05:  Target DN:cn=Lou Smith, ou=Recruiting, ou=HR,
ou=Americas, o=IMC, c=US.
1999/08/03::10:59:05:  Result: Deleted duplicated target entry which was created
later than the change entry. Apply the change entry again.

1999/08/03::10:59:06:  ************ Conflict Resolution Message ************
1999/08/03::10:59:06:  Conflict reason: Attempted to delete a non-existent
entry.
1999/08/03::10:59:06:  Change number:1365.
1999/08/03::10:59:06:  Supplier:eastlab-sun.
1999/08/03::10:59:06:  Change type:Delete.
1999/08/03::10:59:06:  Target DN:cn=Lou
Smith,ou=recruiting,ou=hr,ou=americas,o=imc,c=us.
1999/08/03::10:59:06:  Result: Change moved to low priority queue after failing
on 10th retry.
```

The Replication Process

This section describes how the automated replication process adds, deletes, and modifies entries, and how it modifies DNs and RDNs. It covers topics in the following subsections:

- Adding a New Entry
- Deleting an Entry
- Modifying an Entry
- Modifying a Relative Distinguished Name
- Modifying a Distinguished Name

Adding a New Entry

When it successfully adds a new entry to a consumer, the replication server follows this change application process:

1. The replication server looks in the consumer directory for the DN of parent of the target entry. Specifically, it does this by looking for a **global unique identifier (GUID)** assigned to that parent's DN.
2. If the parent entry exists, the replication server composes a DN for the new entry and places the new entry under its parent in the consumer directory. It then places the change in the Purge Queue.

If the change is not successfully applied on the first try:

The replication server places the new change in the Retry Queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change is not successfully applied on *all but the last* retry:

The replication server keeps the change in the Retry Queue, decrements the number of retries, and repeats the change application process.

If the change is not successfully applied on the last retry:

The replication server checks to see if the change is a duplicate of the target entry.

- **If the change is a duplicate entry**:

  The replication server applies the following conflict resolution rules:

  a. Older creation time stamp wins.

  b. If both entries have the same creation time stamp, the entry with the smaller GUID wins.

  If the change entry wins, then the target entry is removed, the change is applied, and the change entry is placed in the Purge Queue.

  If the target entry wins, then the change entry is placed in the Purge Queue.

- **If the change is not a duplicate entry:**

  The replication server places the change in the Human Intervention Queue, and repeats the change application process at specified intervals.

If the change is not successfully applied after it has been placed in the Human Intervention Queue:

> The replication server keeps the change in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator.

Deleting an Entry

When it deletes an entry from a consumer, the replication server follows this change application process:

1. The 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, the replication server deletes it. It then places the change in the Purge Queue.

If the change is not successfully applied on the first try:

> The replication server places the change in the Retry Queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change is not successfully applied on *all but the last* retry:

> The replication server keeps the change in the Retry Queue, decrements the number of retries, and repeats the change application process.

If the change is not successfully applied on the last retry:

The replication server places the change in the Human Intervention Queue and repeats the change application process at specified intervals.

If the change is not successfully applied after it has been placed in the Human Intervention Queue:

The replication server keeps the change in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator.

Modifying an Entry

When it modifies an entry in a consumer, the replication server follows this change application process:

1. The 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, the replication server compares each attribute in the change entry with each attribute in the target entry.
3. The replication server then applies the following conflict resolution rules:
   a. Most recent modify time wins.
   b. Most recent version of the attribute wins.
   c. The modified attribute on the host whose name is closest to the beginning of the alphabet wins.
4. The replication server applies the filtered modification, and places the change entry in the Purge Queue.

If the change is not successfully applied on the first try:

The replication server places the change in the Retry Queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change is not successfully applied on *all but the last* retry:

The replication server keeps the change in the Retry Queue, decrements the number of retries, and repeats the change application process.

If the change is *not* successfully applied by the last retry:

The replication server places the change in the Human Intervention Queue and repeats the change application process at specified intervals.

If the change is not successfully applied after it has been placed in the Human Intervention Queue:

The replication server keeps the change in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator.

Modifying a Relative Distinguished Name

When it modifies the RDN of an entry in a consumer, the replication server follows this change application process:

1. The 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 replication server modifies the RDN of that entry and places the change entry in the Purge Queue.

If the change is not successfully applied on the first try:

The replication server places the change in the Retry Queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change is not successfully applied on *all but the last* retry:

The replication server keeps the change in the Retry Queue, decrements the number of retries, and repeats the change application process.

If the change is not successfully applied on the last retry:

The replication server places the change in the Human Intervention Queue and checks to see if it is a duplicate of the target entry.

- **If the change is a duplicate entry**:

  The replication server applies the following conflict resolution rules:

  a. Older creation time stamp wins.

  b. If both entries have the same creation time stamp, then the entry with the smaller GUID wins.

  If the change entry wins, then the target entry is removed, the change is applied, and the change entry is placed in the Purge Queue.

  If the target entry wins, then the change entry is placed in the Purge Queue.

- **If the change is not a duplicate entry:**

  The replication server places the change in the Human Intervention Queue, and repeats the change application process at specified intervals.

If the change is not successfully applied after it has been placed in the Human Intervention Queue:

> The replication server keeps the change in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator.

Modifying a Distinguished Name

When it modifies the DN of an entry in a consumer, the replication server follows this change application process:

1. The replication server looks in the consumer for the DN with a GUID that matches the GUID in the change entry.

   The 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 replication server modifies the DN of that entry and places the change entry in the Purge Queue.

If the change is not successfully applied on the first try:

> The replication server places the change in the Retry Queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change is not successfully applied on *all but the last* retry:

> The replication server keeps the change in the Retry Queue, decrements the number of retries, and repeats the change application process.

If the change is *not* successfully applied by the last retry:

> The replication server places the change in the Human Intervention Queue and checks to see if it is a duplicate of the target entry.

- **If the change is a duplicate entry**:

  The replication server applies the following conflict resolution rules:

  a. Older creation time stamp wins.

  b. If both entries have the same creation time stamp, then the entry with the smaller GUID wins.

  If the change entry wins, then the target entry is removed, the change is applied, and the change entry is placed in the Purge Queue.

  If the target entry wins, then the change entry is placed in the Purge queue.

- **If the change is not a duplicate entry:**

  The replication server places the change in the Human Intervention Queue, and repeats the change application process at specified intervals.

If the change is not successfully applied after it has been placed in the Human Intervention Queue:

The replication server keeps the change in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator.

11

Managing National Language Support (NLS)

Oracle Internet Directory National Language Support (NLS) enables you 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 NLS 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.

Oracle Corporation recommends that, prior to configuring NLS, you review the conceptual discussion in "National Language Support" on page 2-26.

This chapter covers topics in the following sections:

- The NLS_LANG Environment Variable
- Using NLS with LDIF Files
- Using NLS with Command Line Tools
- Setting NLS_LANG in the Client Environment
- Using NLS 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 NLS 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, 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. For a complete list of languages, see *Oracle8i National Language Support Guide*.
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. For a complete list of territories, see *Oracle8i National Language Support Guide*
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 the installation or user's guide. For a complete list of character sets, see *Oracle8i National Language Support Guide*.

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 value will be parsed as a language name.

You can set NLS_LANG as an environment variable at the command line. For example, you could specify the value of NLS_LANG by entering either of the following lines at the prompt:

- `AMERICAN_AMERICA.UTF8`
- `JAPANESE_JAPAN.UTF8`

Using NLS with LDIF Files

See Also: "Using LDAP Data Interchange Format (LDIF)" 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 subset of UTF8, 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, including UTF-8 strings, can be present in the LDIF files as attribute values or given as keyboard input. Multibyte strings can be encoded in their native character set or in UTF-8. Multibyte strings can also be a BASE64 encoded representation of either the native or the UTF8 string.

Consider the following cases, each of which is described more fully below:

- CASE 1: Native Strings (Non-UTF8)
- CASE 2: UTF-8 Strings
- CASE 3: BASE64 Encoding of UTF8 Strings
- CASE 4: BASE64 Encoding of 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-UTF8)

Use the `-E` argument in the command line tools, ldifwrite, and bulkmodify. Use the -encode argument in the bulkload and bulkdelete tools.

For example:

```
ldapsearch -h my_host -p 389 -E ".ZHS16GBK" -b base_DN  -s base  'objectclass=*'
```

This example converts simplified Chinese native strings to UTF-8. The baseDN can be a simplified Chinese string.

CASE 2: UTF-8 Strings

No conversion required.

CASE 3: BASE64 Encoding of UTF8 Strings

You do not need to use 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 UTF8 strings to UTF8 strings.

CASE 4: BASE64 Encoding of 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 native strings and the native strings are then converted to UTF8 strings.

> **Note:** In any given input file, all language set values should be for the same language set.

Using NLS with Command Line Tools

Command line tools are non-Java clients that support reading keyboard input or LDIF file input in the following ways:

- ASCII characters only from keyboard input or LDIF file
- Non-ASCII input from keyboard or LDIF file (native language character set)
- LDIF file containing BASE64 encoded values (of UTF-8 or native language character set)

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.

Specifying the -E Argument When Using Each Tool

Specifying the `-E` argument ensures that proper character set conversion can occur from the character set you specify for the `-E` argument (`-E ".`*`xxxx`*`"`) to the .UTF8 character set.

The command line tools use the `-E` argument to process the input in the character set specified for the `-E` argument. They use the NLS_LANG environment variable to process the output in the character set specified by `NLS_LANG`.

For example, to add an LDIF file encoded in the .ZHS16GBK (simplified Chinese) character set by using ldapadd, you would type:

```
ldapadd -h myhost -p 389 -E ".ZHS16GBK" -f my_ldif_file
```

In this example, the characters are converted from `".ZHS16GBK"` (simplified Chinese character set) to "`.UTF8`" (UTF-8 character set) before they are sent across the wire to the LDAP server.

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.

Examples: Using the -E Argument with Command Line Tools

The following table provides examples of how to use this additional 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 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`
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`

Tool	Example
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,o=acme,c=us -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 you specify for the NLS_LANG environment variable.

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.

Using NLS with Bulk Tools

Oracle Internet Directory ensures that the reading and writing of text data from and to the LDIF files are done in UTF-8 encoding as specified by LDAP.

This section provides an example of the argument you use for each of the following bulk tools:

- bulkload
- ldifwrite
- bulkdelete
- bulkmodify

See Also: "Importing an LDIF File by Using bulkload" on page 7-26 for a list of arguments for each bulk tool

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" -check -generate -load
my_ldif_file
```

ldifwrite

The ldifwrite utility always writes BASE64 encoded values for multibyte strings.

The BASE64 encoding could be of the UTF8 strings as they are stored in the database, 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 a BASE64 encoded value of UTF-8 strings.

To reload this LDIF file into the directory by using ldapaddmt, use the following syntax:

```
ldapaddmt -h host -p port -f output_file
```

In the above case, the `-E` argument is not required because the decoded BASE64 strings are already in UTF8 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 output_file
```

In the above case the `-E` argument is required because the decoded BASE64 strings are simplified Chinese, which need to be converted to UTF8 strings before being sent to the server.

bulkdelete

Add to the command the argument `-encode ".character_set"`.

For example:

```
bulkdelete.sh -connect net_service_name -encode ".ZHS16GBK" -base
ou=manufacturing,o=acme,c=us -size 100
```

In this case the value for the `-base` option could be in the `ZHS16GBK` native character set, that is, simplified Chinese.

bulkmodify

Add to the command the argument `-E ".character_set"`.

For example:

```
bulkmodify -c net_service_name -E ".ZHS16GBK" -b ou=manufacturing,o=acme,c=us -r
title -v Foreman -f filter -s 100
```

The above values for the `-b`, `-v,` and `-f arguments` could be specified in native character set.

Part III

Deploying Oracle Internet Directory

Part III discusses deployment considerations. Specific chapters are:

12

Capacity Planning

This chapter discusses topics in the following sections:

- About Capacity Planning
- Getting to Know Directory Usage Patterns: Acme Corporation
- I/O Subsystem Requirements
- Memory Requirements
- Network Requirements
- CPU Requirements
- Summary of Capacity Plan for Acme Corporation

About Capacity Planning

Capacity planning is the process of assessing applications' directory access requirements and ensuring that the Oracle Internet Directory has adequate machine resources to service requests at an acceptable rate. Assuming that the Oracle Internet Directory and the corresponding Oracle8*i* database are running on the same machine, the following 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—in other words, 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.

Useful Definitions

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: Acme Corporation

The ability to assess the potential load on Oracle Internet Directory is very important for getting an accurate capacity plan. Let us take a typical email messaging software in a big multinational company called Acme Corporation. The email messaging software 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

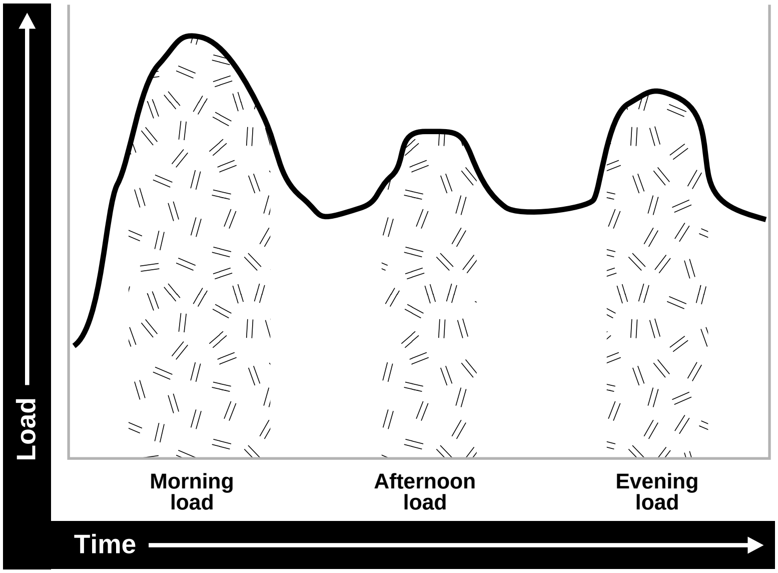
The above assumptions will yield a directory population of about one million entries. Given the user population and the directory population, let us then analyze usage patterns so that we can derive performance requirements from them. A typical user tends to send an average of 10 emails per day and receives 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 12–1.

Figure 12–1 Usage Analysis of Current Email System



The email system (and Oracle Internet Directory) is stressed at its peak in the mornings. There are other usage peaks as well—one close to lunch time, and one near the end of business day. But it is in the mornings that the Oracle Internet Directory is stressed the most.

Let us assume that 90% 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

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 13. 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 rest of the section on deriving sizing estimates for the disks themselves.

Rough Estimates of Disk Space Requirements

Table 12–1 can be used to derive a rough estimate of the overall disk requirement:

Table 12–1 Estimating Disk Space Requirements

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 Table 12–1 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 table above 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 Oracle 8*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 Directory Information Tree (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 distinguished name 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 Oracle8*i* database for various book-keeping purposes. Typically, its size remains constant at about 300MB

In this section, we present simple arithmetic procedures to determine the size requirements of each of the tablespaces shown above. All of the size calculations will be based on the following variables:

Table 12–2 Variables for Size Calculations

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 that we wish 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% overhead. The minimum value for this variable is 1.

Table 12–2 Variables for Size Calculations

Variable Name	Description
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% 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% factor of safety. The minimum value for this variable is 1.

Using the variables shown in Table 12–2, the size of individual tablespaces can be calculated as follows:

Table 12–3 Calculating Tablespace Sizes

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 Table 12–3, 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 in 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_ATTR_STORE	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% factor of safety, this would jump to 10.4GB. If all of the data is being loaded in bulk, then the bulkloader tool of Oracle Internet Directory would require an additional 50% 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

Oracle Internet Directory is a database application. Memory is used for a number of distinct tasks by any database application. 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% 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 13 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	512MB
medium	600,000 to 2,000,000	1GB
large	greater than 2,000,000	2GB

Going back to our example of Acme Corporation, the number of entries in the directory are close to 1,000,000 (1 million). We recommend choosing the 2GB 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 machine running the Oracle Directory Server is only accessible through a 10 Mbps network (10-Base-T switched ethernet), and we have only 60% 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:

% 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 82 directory operations per second. Assuming that each operation results in a transfer of 2KB of data on the network, this would imply that we should have a 100 Mbps network or at least 60% 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

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 a mail 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 machine 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 (`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.

Since Oracle Internet Directory makes efficient use of multiple CPUs on an SMP machine, we chose to categorize machines 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 machine, 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 Oracle8*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% 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% 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 machines. We then correlated this to SPECint_rate95 baseline number of the machines. 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 = 6.0 * (concurrent base search operations)
```

For example, if we need a machine that is capable of handling 50 concurrent base search operations before saturating the CPU, we would require a machine 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% to 30% (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%, our preliminary estimate of the CPU requirements would be:

```
SPECint_rate95 baseline=6.0*100*(1 + 0.2*1.5)*1.2 = 600*1.3*1.2 = 936
```

Looking at the available systems from the SPEC web site (`www.spec.org`) we can see that the following machine configurations would be the smallest configurations that should be considered. Table 12–4 shows some of the machines that Acme Corporation can consider for using for Oracle Internet Directory.

Table 12–4 Machine Configurations

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.

13

Tuning

This chapter gives guidelines for tuning an Oracle Internet Directory installation. Topics are discussed in the following sections:

Introduction

Once capacity planning as described in Chapter 12 is complete, and the necessary hardware acquired, it is important to perform some test runs to figure out if the hardware and software combination is yielding the desired levels of performance. 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
NT Performance Monitor	Gives a customized view of the events in the system
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 Oracle8*i*:

- UTLBSTAT.SQL and UTLESTAT.SQL
- The ANALYZE function in the DBMS_STATS package

> **See Also:**
>
> - *Oracle8i Reference* for information about UTLBSTAT.SQL and UTLESTAT.SQL
> - *Oracle8i 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.

CPU Usage Tuning

The CPU is perhaps the most important resource available for any software. While Chapter 12 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. Tuning of CPU resources should be considered if either of the following cases are true:

- At peak loads the CPU is 100% 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% of the CPU resources are consumed by Oracle Internet Directory processes, and the remaining (about 25% to 30%) 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% user and 15% system time.

This section discusses topics in the following subsections:

- 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. The following 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 10 server processes created.
- Each server process will spawn 15 worker threads which will do the actual work.
- Each server process will also maintain a pool of 16 database connections (15+1) which will be shared among the worker threads.

Tuning Oracle Internet Directory Processes When CPU Is 100% Utilized

If the CPU usage of the system is at 100%, further tuning of the Oracle Internet Directory processes should be considered if *all* of the following conditions are met:

- At peak loads, Oracle Internet Directory processes consume more than 70% of all available CPU resources.
- At peak loads, the overall percentage of time spent in the 'system' or 'kernel' space is greater than 20%, and the percentage of time spent in the 'user' time is less than 80%.

This condition indicates that the system is over-configured with the number of Oracle Internet Directory server processes and database connections. 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% or higher
- System time: 15% or lower

Tuning Oracle Internet Directory Processes When CPU Is Under-Utilized

If the CPU usage at peak loads is not at 100% and the system is idle for a large percentage of the time (that is, more than 5%), 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% and the system and user time are split up as follows:

- User time: 85% or higher
- System time: 15% or lower

Tuning CPU for Oracle Foreground Processes

Tuning of CPU resources for Oracle Foreground processes should be considered only if *all* of the following conditions are met:

- The CPU usage is close to 100% at peak loads.
- Oracle foreground processes consume more than 30% 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%.
- There are more than two CPUs on the computer.
- Oracle Internet Directory processes consume around 70% to 75% of the CPU resources.
- The database processes consume around 25% to 30% 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 in 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 Oracle8*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 Oracle8*i* database. The System Global Area (SGA) of the back-end database must be made as large as possible while leaving room for Oracle Internet Directory and Oracle processes to operate their private stacks and heaps. The following section provides some details on determining various components of the SGA.

This section discusses topics in the following subsections:

- Tuning the System Global Area (SGA) for Oracle8i
- Other Alternatives for a Memory-Constrained System

Tuning the System Global Area (SGA) for Oracle8*i*

The SGA should be sized based on the available physical memory on the system running Oracle8*i*.

> **See Also:** *Oracle8i Designing and Tuning for Performance* for more information on determining appropriate sizes for the SGA to ensure that the SGA size does not cause increased paging swapping activity, which is very detrimental to performance

Once the available size of the SGA is determined, two primary tuning items need to be considered: the size of the shared pool, and the 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% of the total SGA, use 30% of the total SGA instead.

Divide 60% 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 Internet Directory LDAP server on the same computer, one can put the database on a different computer.

Disk Tuning

Balancing Disk IO is an important consideration in overall RDBMS (and hence Oracle Internet Directory performance). 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

> **See Also:** See *Oracle8i Designing and Tuning for Performance* for general information on balancing and tuning disk I/O

The remainder of this section discusses topics in the following subsections:

- 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 13-9 for more discussion on logical disks

Separate the following:

- OLTS_ATTRSTORE and OLTS_IND_ATTRSTORE

  Separating the attribute store table from it's index
- OLTS_CT_DN and OLTS_IND_CT_DN

  Separating the DN catalog from it's 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 IO. (Two logical drives on the same physical disk media do not really provide the same combined IO 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 IO capacity of that logical drive, but the tablespace locations considered above remain applicable when considering different logical drives of a volume manager, for instance.

Database Tuning

This section describes the other tunables 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:

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 in the following subsections:

- Required Parameters
- Parameters Dependent on Oracle Internet Directory Server Configuration
- SGA Parameters Dependent on Hardware Resources

Required Parameters

OPEN_CURSORS=100

Oracle8*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, # WORKERS, etc. The value of 100 is sufficient for any size DIT.

Parameters Dependent on Oracle Internet Directory Server Configuration

SESSIONS = processes = (# OID server processes per instance) x

(# DB Connections per server + 1) x

(# of OID instances) + 20

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 MTS

Depending on the total number of concurrent database connections required (as determined above, enabling MTS may help balance overall system load better. If the total number of concurrent database connections required is over 300, then configure MTS. 1 shared server should be configured for every 10 database connections required.

> **Note:** The number of requires concurrent database connections depends on the hardware selected. See *Oracle8i Designing and Tuning for Performance* for more exact guidelines.

SGA Parameters Dependent on Hardware Resources

The main parameters that contribute to the SGA are discussed in "Memory Tuning" on page 13-7. 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 LGWR 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.
- Make sure that 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
- There are enough rollback segments in the database
- The schema associated with the 'ODS' user is 'ANALYZED'

14

High Availability And Failover

This appendix discusses the high availability and failover features and deployment guidelines for Oracle Internet Directory. It covers the following topics:

Introduction

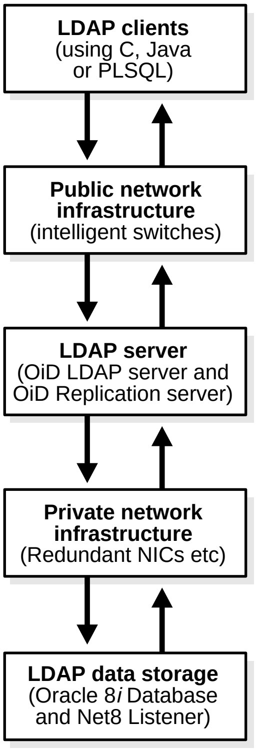
Oracle Internet Directory is designed to address the deployment needs of mission critical applications requiring a high degree of system availability. To achieve a high degree of availability, all components in the system must facilitate redundancy, and all interfaces must facilitate failure recognition 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 14-3. Typically, it is not necessary to employ every failover capability in every component. This appendix describes the availability and failover features of various components in the Oracle Internet Directory / Oracle8*i* technology stack and provides guidelines for exploiting them optimally for typical directory deployment.

The Oracle Internet Directory/Oracle8*i* Technology Stack

Figure 14–1 gives an overview of the various components of the Oracle Internet Directory stack.

Figure 14–1 Oracle Internet Directory/Oracle8i 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 could failover to alternate LDAP servers in case the primary LDAP 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.

Alternate Server List from User Input

The clients can be designed to take input from the user on the list of alternate LDAP 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 `AlternateServers`. 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 LDAP 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:

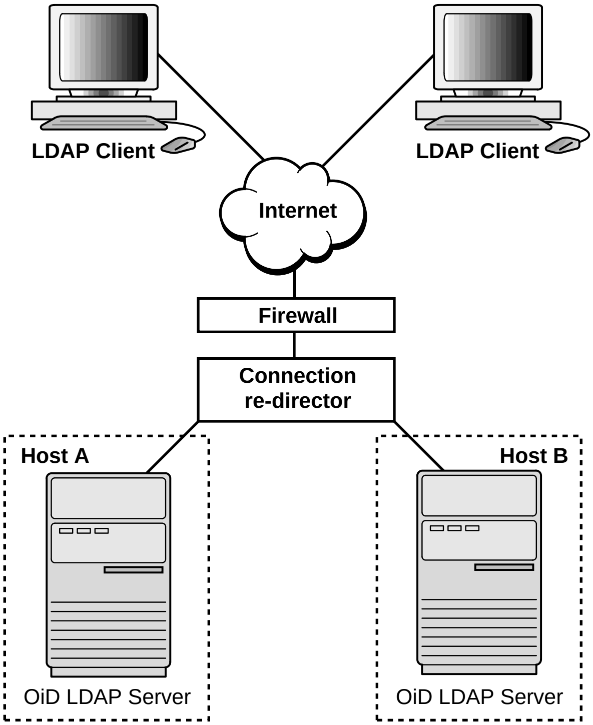
- "Managing Attributes by Using Oracle Directory Manager" on page 6-18
- "Managing Attributes by Using Command Line Tools" on page 6-29

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 machines running the Oracle Internet Directory LDAP server and an intelligent TCP/IP level connection re-director. In both cases, there would be more than one machine serving LDAP requests so that failure of one LDAP server machine would not affect availability. Figure 14–2 illustrates a typical Internet deployment of Oracle Internet Directory with network-level failover enabled.

Figure 14–2 Network-Level Failover



In Figure 14–2, the Oracle Internet Directory 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.

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 machines as shown in Figure 14–2. They can also detect machine failures and stop routing requests to the failed machine. This feature guarantees that new connections from clients will not be routed to a failed machine. When a machine 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

Multi-master 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 multi-master configuration. A replica node is recommended for each region that is separated from others by relatively low speed or low bandwidth network segment. Such a configuration, while allowing speedy directory access to the clients in the same region, will also serve as a failover arrangement during regional failures elsewhere.

Failover Options in the Private Network Infrastructure

We define the Private Network Infrastructure as 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, it is highly recommended 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 get dedicated bandwidth with the help of a network switch. Because Oracle Internet Directory critically depends on the Private Network Infrastructure for its communications, adequate precautions must be taken 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) to protect against network adapter failures
- Redundant links to protect against link failure

IP Address Takeover (IPAT)

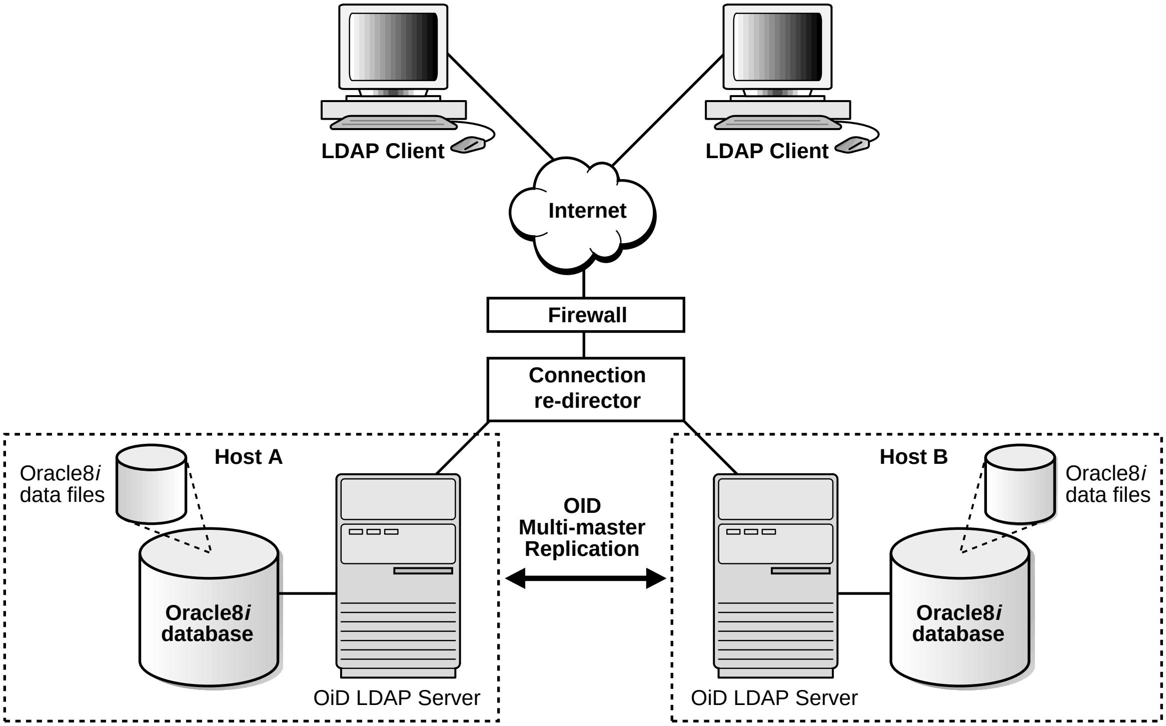
IP address takeover feature is available on many commercial clusters. This feature protects an installation against failures of the Network Interface Cards (NICs). In order to make this mechanism work, installations must have two NICs for each IP address assigned to a server. Both the NICs must be connected 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 machine to a server machine. If installations want to take such precautions, they should use NICs and hubs/switches that come with the capability to use redundant links in case of a link level failure.

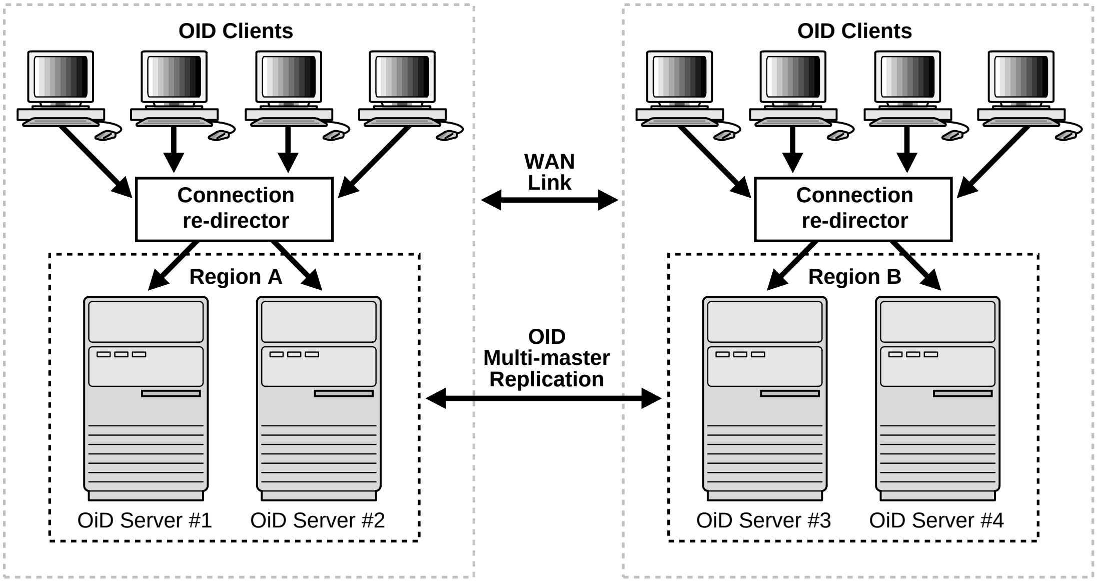
Deployment Examples

Figure 14–3 Deployment Example (Two Oracle Internet Directory Nodes in Replication)



In Figure 14–3, the Oracle8*i* database and LDAP server are co-resident on the same machine. Changes made on one LDAP server instance get reflected on the second LDAP server instance through multi-master replication. When a failure of the LDAP server or database server on a particular node occurs, it is elevated to a machine failure so that the connection redirector will stop handing off connections to the machine on which there was a failure.

Figure 14–4 Deployment Example 2



As the example in Figure 14–4 illustrates, each of the regions can be setup 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.

Part IV

Appendixes

Part IV contains the following appendixes:

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 in the following sections:

- Using LDAP Data Interchange Format (LDIF)
- Using Command Line Tools
- Using Bulk Tools
- Using the Catalog Management Tool
- Using the OID Database Password Utility

Using LDAP Data Interchange Format (LDIF)

The standardized file format for directory entries is as follows:

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.

Example 14–1 LDIF File Entry for an Employee

```
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: organizational person
objectClass: person
objectClass: top
```

In Example 14–1, 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.

Example 14–2 LDIF File Entry for an Organization

```
dn: o=Acme,c=US
o: Oracle
ou: Financial Applications
objectClass: organization
objectClass: top
```

LDIF Formatting Notes

The following list of LDIF formatting rules 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, you can use Oracle Directory Manager. See "Viewing Properties of Object Classes" on page 6-9.

- Non-printing characters and tabs are represented in attribute values by base-64 encoding.
- The entries in your data 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 "Further Reading" on page 2-33 for a complete list of LDIF formatting rules
  - "Using NLS with LDIF Files" on page 11-3.

Using Command Line Tools

This section tells you how to use the following tools:

- ldapsearch
- ldapbind
- ldapadd
- ldapaddmt
- ldapmodify
- ldapmodifymt
- ldapdelete
- ldapcompare
- ldapmoddn

ldapsearch

The ldapsearch command line tool searches for and retrieves specific entries in the directory.

To run ldapsearch at the command line, use this syntax:

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

The filter format should be compliant with RFC-2254. For further information about this standard, search for the standard at: http://www.ietf.org/rfc/rfc2254.txt

Separate attributes with a space.

If you do not list any attributes, all attributes are retrieved.

Mandatory Arguments	Descriptions
-b *basedn*	Specifies base dn for search
-s *scope*	Specifies search scope: base, one, or sub.

Optional Arguments	Descriptions
-A	Retrieves attribute names only (no values)
-a *deref*	Specifies alias dereferencing: never, always, search, or find

Optional Arguments	Descriptions
-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.
-d *debug level*	Sets debugging level to the level specified (see Table 5–5 on page 5-21)
-E "*character_set*"	Specifies native character set encoding. See Chapter 11.
-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 machine. *ldaphost* can be a machine 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 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)
-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. Each of these examples searches on port 389 of host sun1, and searches the whole subtree starting from the DN "ou=hr,o=acme,c=us".

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree "objectclass=*"
```

This search will find all entries with any value for the objectclass attribute.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"objectclass=orcle*"
```

This search will find all entries that have orcle 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=orcle*)(cn=foo*))"
```

This search returns entries where the objectclass attribute begins with *orcle* and cn begins with foo.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree "(!(cn=foo))"
```

This search returns entries where the common name (cn) is not *foo*.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"(|(cn=foo*)(sn=bar*))"
```

This search returns entries where cn begins with foo or sn begins with bar.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"employeenumber<=10000"
```

This search returns entries where employeenumber is less than or equal to 10000.

ldapbind

Use the ldapbind command line tool to see whether you can authenticate a client to a server.

To run ldapbind, type at the command line:

```
ldapbind [options]
```

Optional Arguments	Descriptions
-D *binddn*	When authenticating to the directory, specifies doing so as the entry specified in *binddn*. Use this with the `-w` *bindpassword* option.
-E ".*character_set*"	Specifies native character set encoding. See Chapter 11.
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local machine. *ldaphost* can be a machine name or an IP address.
-n	Shows what would occur without actually performing the operation
-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)
-w *password*	Provides the password required to connect
-W *wallet_location*	Specifies wallet location (required for one-way or two-way SSL connections)

ldapadd

Use the ldapadd command line tool 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" on page A-11.

> **See Also:** "Adding Configuration Set Entries by Using ldapadd" on page 5-12 for an explanation of using ldapadd to configure a server with an input file

To run the ldapadd command, type at the command line:

```
ldapadd options -f filename
```

where *`filename`* is the name of an LDIF file written with the specifications explained in detail later in this section.

Optional Arguments	Descriptions
-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 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` *bindpassword* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 11.
-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 "Using LDAP Data Interchange Format (LDIF)" on page A-2.
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local machine. *ldaphost* can be a machine 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.
-n	Shows what would occur without actually performing the operation
-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
-w *password*	Provides the password required to connect
-W *wallet_location*	Specifies wallet location (required for one-way or two-way SSL connections)

ldapaddmt

ldapaddmt is like ldapadd: it adds 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.

To run ldapaddmt, use this command 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 detail later in this section.

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
```

Increasing the number of concurrent threads improves the rate at which LDIF entries are created, but consumes more system resources.

Optional Arguments	Descriptions
-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 11
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local machine. *ldaphost* can be a machine 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.
-n	Shows what would occur without actually performing the operation.
-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
-w *password*	Provides the password required to connect
-W *wallet_location*	Specifies wallet location (required for one-way or two-way SSL connections)

ldapmodify

The ldapmodify tool acts on attributes.

To run the ldapmodify command, use this command syntax:

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

where *filename* is the name of an LDIF file written with the specifications explained in detail later in this section.

The list of arguments in the following table is not exhaustive.

Optional Arguments	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` *bindpassword* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 11.
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local machine. *ldaphost* can be a machine name or an IP address.
-n	Shows what would occur without actually performing the operation.
-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

Optional Arguments	Description
-w *bindpassword*	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)

To run `modify`, `delete`, and `modifyrdn` operations using the `-f` flag, use LDIF for the input file format (see "Using LDAP Data Interchange Format (LDIF)" on page A-2) with the specifications noted below:

Always separate entries with a blank line.

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 "Using LDAP Data Interchange Format (LDIF)" 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 immediately below:

- **Add attribute values**. This option to changetype modify adds more values to an existing multi-valued attribute. If the attribute does not exist, it will add 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:delete
  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," 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 RDN (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
  ```

ldapmodifymt

Use the ldapmodifymt command line tool to modify several entries concurrently.

To run ldapmodifymt, use this command syntax:

```
ldapmodifymt -T number_of_threads [options] -f filename
```

where *filename* is the name of an LDIF file written with the specifications explained in detail later in this section.

For example:

```
ldapmodifymt -T 5 -h node1 -p 3000 -f myentries.ldif
```

Optional Arguments	Descriptions
-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.)
-D *binddn*	When authenticating to the directory, specifies doing so as the entry is specified in binddn. Use this with the `-w` *bindpassword* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 11.
-h *ldaphost*	Tells ldapmodify to connect to *ldaphost*, rather than to the default directory. *ldaphost* can be an IP address.
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local machine. *ldaphost* can be a machine name or an IP address.
-n	Shows what would occur without actually performing the operation.
-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
-w *bindpassword*	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)

See Also: "ldapmodify" on page A-11 for additional formatting specifications used by ldapmodifymt

ldapdelete

The ldapdelete command line tool removes entire entries from the directory that you specify in the command line.

To delete an entry by using ldapdelete, use this command syntax:

```
ldapdelete [options] "entry_DN"
```

The following example uses port 389 on a host named myhost.

```
ldapdelete -p 389 -h myhost ou=EuroSInet Suite, o=IMC, c=US”
```

Optional Arguments	Descriptions
-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 5-21.
-E "*character_set*"	Specifies native character set encoding. See Chapter 11.
-f *filename*	Specifies the input filename
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local machine. *ldaphost* can be a machine 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.
-n	Shows what would be done, but doesn’t actually delete
-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

Optional Arguments	Descriptions
-v	Specifies verbose mode
-w *password*	Provides the password required to connect.
-W *wallet_location*	Specifies wallet location (required for one-way or two-way SSL connections)

ldapcompare

The ldapcompare command line tool matches attribute values you specify in the command line with the attribute values in the directory entry.

To run ldapcompare, use this command syntax:

```
ldapcompare [options]
```

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	Descriptions
-a *attribute name*	Specifies the attribute on which to perform the compare
-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	Descriptions
-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 5-21.
-E "*character_set*"	Specifies native character set encoding. See Chapter 11.
-f *filename*	Specifies the input filename
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local machine. *ldaphost* can be a machine name or an IP address.
-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 Arguments	Descriptions
-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
-w *password*	Provides the password required to connect
-W *wallet_location*	Specifies wallet location (required for one-way or two-way SSL connections)

ldapmoddn

The ldapmoddn command line tool modifies the DN or RDN of an entry.

To run ldapmoddn, use the following syntax:

```
ldapmoddn [options]
```

The following example uses ldapmoddn to modify the RDN component of a DN from "cn=dcpl" to " cn=thanh mai". It uses port 389, and a host named myhost.

```
ldapmoddn -p 389 -h myhost -b "cn=dcpl,dc=Americas,dc=imc,dc=com" -R "cn=thanh
mai"
```

Mandatory Argument	Description
-b *basedn*	Specifies DN of the entry to be moved

Optional Arguments	Descriptions
-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 11.
-f *filename*	Specifies the input filename
-h *ldaphost*	Specifies name of the host node of the directory server
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local machine. *ldaphost* can be a machine name or an IP address.

Optional Arguments	Descriptions
-N *newparent*	Specifies new parent of the RDN
-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)
-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
-w *password*	Provides the password required to connect.
-W *wallet_location*	Specifies wallet location (required for one-way or two-way SSL connections)

Using Bulk Tools

This section tells you how to use the following bulk tools:

- bulkload
- ldifwrite
- bulkmodify
- bulkdelete

bulkload

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. The bulkload tool expects the input file to be in the LDAP Data Interchange Format (LDIF).

> **See Also:** "Using LDAP Data Interchange Format (LDIF)" on page A-2.

The bulkload tool syntax is:

```
bulkload.sh -connect net_service_name [-check] [-generate] [-load]
   [-restore] absolute_path_to_ldif.file
```

Mandatory Argument	Description
connect *net_service_name*	Connects to the database using the net service name defined in the tnsnames.ora file

Optional Arguments	Descriptions
check	Checks LDAP schema for inconsistencies and for existence of duplicate DNs in the data file
-encode "*character_set*"	Specifies native character set encoding. See Chapter 11.
generate	Creates files suitable for loading into Oracle Internet Directory
load	Loads files resulting from generate phase into specified database

Optional Arguments	Descriptions
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 Oracle Internet Directory processes are not running.

> **See Also:** Chapter 5 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 machines with the identical SQL*Loader file. Do this only when creating a new replica node.

> **See Also:** "Step 7: Start the Replication Servers on All the Nodes" on page 10-16.

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).

ldifwrite

The ldifwrite command line tool enables you to convert all or part of the information residing in an Oracle Internet Directory to LDIF. This makes that information available for loading into a new node in a replicated directory or into another node for backup storage. The ldifwrite tool performs a subtree search, including all entries below the specified DN, including the DN itself.

The ldifwrite tool syntax is as follows:

```
ldifwrite -c net_service_name -b base_DN -f filename
```

Mandatory Arguments	Descriptions
-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.
-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 Chapter 11.

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`.

> **See Also:** "ldifwrite" on page 11-8 for information on specifying the `-E` option and using National Language Support with ldifwrite

bulkmodify

Use the bulkmodify tool 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 LDAP server and 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 the 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 syntax is as follows:

```
bulkmodify -c net_service_name -b base_dn {-a|-r} attr_name -v att_value [-f
filter] [-s size]
```

Mandatory Arguments	Descriptions
-c *net_service_name*	Specifies the net service name of the directory database to connect to
-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 Arguments	Descriptions
-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*"	Native character set encoding. See Chapter 11.

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:

```
-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.

bulkdelete

Use the bulkdelete command line tool for deleting a subtree efficiently. It can be used when both an LDAP server and 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. The administrator must restrict LDAP activity against the subtree during deletion.

The bulkdelete tool syntax is as follows:

```
bulkdelete.sh -connect net_service_name -base "base_dn" -size number_of_entries
```

The script includes the following arguments:

Mandatory Arguments	Descriptions
- connect *net_service_name*	Specifies the net service name to connect to the directory database
- base "*base_dn*"	Specifies the base DN of the subtree to be deleted

Optional Arguments	Descriptions
-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

Using the Catalog Management Tool

Before running the Catalog Management tool, unset the LANG variable. After you finish running Catalog Management tool, set the LANG variable back to its original value.

To unset LANG:

- Using Korn shell:

```
UNSET LANG
```

- Using C shell:

```
UNSETENV LANG
```

The Catalog Management tool syntax is:

```
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

Optional Arguments	Descriptions
- 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
```

After you finish running the Catalog Management tool, set the LANG variable back to its original value.

To set LANG:

- Using Korn shell:

  ```
  SET LANG=appropriate_language, EXPORT LANG
  ```

- Using C shell:

  ```
  SETENV LANG appropriate_language
  ```

To effect the changes after running the Catalog Management tool, stop, then restart, Oracle Directory Server.

> **See Also:** Chapter 5 for instructions on starting and restarting directory servers

Using the OID Database Password Utility

The OID Database Password Utility syntax is as follows:

```
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. Note that none of the passwords is echoed to the screen.

The OID Database Password Utility assumes by default that the local database (as defined by ORACLE_HOME and ORACLE_SID) is the database password being changed. 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.

B

Adding a DSA Using the Database Copy Procedure

This appendix describes the procedure to add a new **DSA** to an existing replicating system 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 10, 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 covers the following 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 platform 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
SQL> connect internal
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=<inst_#> stop
   $ oidctl connect=<net_service_name> server=oidldapd instance=<inst_#> 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=<inst_#> stop
   ```

   Repeat the above procedure on all nodes except the sponsor node. Specify appropriate net service names for the corresponding nodes.

4. Quiesce **Advanced Symmetric Replication (ASR)** on the sponsor node (**Master Definition Site (MDS)**) by running the following script:

   ```
   $ cd $ORACLE_HOME/ldap/admin
   $ sqlplus repadmin/repadmin
   SQL> @ldapsuspend.sql
   ```

   Enter the Oracle global name for the sponsor node 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 Net8 listener on the sponsor node only:

   ```
   $ lsnrctl [listener name] stop   (By default listener name is LISTENER)
   $ sqlplus
   SQL> connect internal
   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: The Oracle installation manual

   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 B-8.) 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
SQL> connect internal
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
SQL> connect internal
SQL> startup
SQL> exit
$ lsnrctl start (By default listener name is LISTENER)
$ exit
```

11. Since the new node is created by using backup database copy of the master definition site (sponsor node), the master definition catalog needs to be dropped. Execute the following script which will drop the definition of master definition site from the ASR catalog on the new node.

```
$ cd $ORACLE_HOME/ldap/admin
$ sqlplus repadmin/repadmin
SQL> @ldapdropmds.sql
```

Important: Specify the global name of the new node when prompted.

12. Execute the following SQL scripts.

```
$ cd $ORACLE_HOME/ldap/admin
$ sqlplus repadmin/repadmin
SQL> @ldapcreindex.sql
SQL> @ldapcoldrm.sql
```

    Specify the Oracle system password and global name of the sponsor and new nodes when prompted.

13. Execute the following SQL script on all nodes *including* the new node and the sponsor node.

```
$ cd $ORACLE_HOME/ldap/admin
$ sqlplus repadmin/repadmin
SQL> @ldapjobs
```

14. Resume ASR activity at the master definition site. Execute the following script:

```
$ cd $ORACLE_HOME/ldap/admin
$ sqlplus repadmin/repadmin
SQL> @ldapresume
```

15. To verify the ASR setup, execute the following script on all the nodes:

```
$ cd ORACLE_HOME/ldap/admin
$ sqlplus repadmin/repadmin
SQL> @ldaplogq
```

    You should get "no rows selected." This command could take two to three minutes to report this result.

16. Execute the following script to modify the Replication related catalogs:

```
$ cd ORACLE_HOME/ldap/admin
$ sqlplus repadmin/repadmin
SQL> @ldapcoldadd.sql
```

    Specify the Oracle system password and global name of the sponsor and new nodes when prompted.

17. At this point start the OID Monitor and ldap server on new and sponsor nodes. (It is assumed that OID Monitor and ldap server are already running on other nodes. If they are shut down, then start up the OID Monitor and ldap servers on the other nodes too.)

18. Update the LDAP replication agreements on all nodes to include the new node.

    ```
    Sample LDIF file:

    dn: orclagreementid=000001, cn=orclreplagreements
    changetype: modify
    add: orcldirreplgroupdsas
    orcldirreplgroupdsas: dsm-sun
    ```

    Save the above command to a file replagg

    ```
    $ ldapmodify -h <hostname> -p  <port-id> -f replagg
    ```

19. 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 ASR-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.

C

Troubleshooting

This appendix explains typical problems that you could encounter while running or installing the Oracle Internet Directory. If you suspect that the Oracle Internet Directory processes are working improperly, or you need information about an error code, refer to the relevant section below.

- Installation Errors
- Administration Error Messages and Causes

Installation Errors

During installation and configuration of the Oracle8*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.

The Oracle Internet Directory application replaces the <parameter> tag seen in some of the messages below with the appropriate run-time value.

This section discusses error messages in the following sections:

- 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 C-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.

%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 OID 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 OID 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 OID 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.

D

Using Oracle Wallet Manager

Security administrators use Oracle Wallet Manager to manage public-key security credentials on Oracle clients and servers. Oracle Wallet Manager is used to create wallets that can be later opened by using either the Oracle Enterprise Login Assistant or the Oracle Wallet Manager.

The topics are covered in the following sections:

Overview

Public-key cryptography requires entities, that want to communicate in a secure manner, to possess certain security credentials. This collection of security credentials is stored in a wallet.

Security credentials consist of a public/private key pair, a "user" certificate, a certificate chain, and "trusted" certificates. Each entity that participates in a public key system must have a public/private key pair. The public key for an entity is published in a user certificate so, for example, other entities that want to send it secure information can encrypt that information with the recipient entity's public key. Another use for a public key is for an entity that receives a communication to validate the sender's organizational affiliation—this is the most typical application in Oracle environments today. Oracle Wallet Manager generates public/private key pairs for clients and servers.

A certificate authority (CA) issues public key certificates. A certificate contains a unique serial number assigned to it by the CA, an algorithm identifier that identifies which algorithm was used to sign that certificate, the name of the CA that issued that certificate, a pair of dates between which the certificate is valid, the certificate user's name, the entity's public key, and the CA's signature.

A trusted certificate, sometimes known as a root key certificate, typically belongs to a third party entity that is trusted to issue certificates. It is obtained in a secure manner and, operationally, does not need to be validated for its authenticity each time it is accessed. A client or a server uses a trusted certificate to validate that an entity is who it claims to be by verifying that entity's certificate. Typically, certificate authorities you trust issue the user certificates. Oracle provides several default trusted certificates, so users do not have to install their own. These trusted certificates also enable servers to perform SSL authentication to clients who have wallets containing only trusted certificates.

Clients and servers use these credentials to access secure services, such as SSL, using public key cryptography. A wallet also represents a storage facility that is location and type transparent once it is opened.

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 certificate authority (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 later opened by using either the Oracle Enterprise Login Assistant or the Oracle Wallet Manager.

Security Concepts

General security concepts and their associated definitions are as follows:

Table D–1 Security Concepts and Definitions

Concept	Definition
Authentication	The recipient of an authenticated message can be certain of the message's origin (its sender). Authentication reduces the possibility that another person has impersonated the sender of the message.
Authorization	The set of privileges available to an authenticated entity.
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. This certificate ensures that the entity's information is correct and that the public key actually belongs to that entity.
Certificate Authority	An application that creates public key certificates with a high level of assurance and trust in this function.
Confidentiality	A function of cryptography. Confidentiality guarantees that only the intended recipient(s) of a message can view the message (decrypt the ciphertext).
Cryptography	The act of writing and deciphering secret code resulting in secure messages.
Decryption	The process of converting the contents of an encrypted message (ciphertext) back into its original readable format (plaintext).
Digital Signature	A digital signature is created when a public key algorithm is used to sign the sender's message with the sender's private key. The digital signature assures that the document is authentic, has not been forged by another entity, has not been altered, and cannot be repudiated by the sender.

Table D–1 Security Concepts and Definitions

Concept	Definition
Encryption	The process of disguising the contents of a message (plaintext) and rendering it unreadable (ciphertext) to anyone but the intended recipient.
Identity	A user who is certified as being the entity it claims to be.
Integrity	The guarantee that the contents of the message received were not altered from the contents of the original message sent.
Non-repudiation	Undeniable proof of the origin, delivery, submission, or transmission of a message.
Public-Key Encryption	The process where the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the message is decrypted by the recipient using the recipient's private key. When public-key encryption is used with symmetric key encryption, DES for example, only the DES key is encrypted with the recipient's public key. The message itself is encrypted with a DES key: this is more efficient.
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. The two numbers are related, but it is mathematically infeasible to derive the private key from 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.
Trusted Certificate	A trusted certificate, sometimes known as a root key certificate, is a third party identity that is trusted. 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. If there are several levels of trusted certificates, a trusted certificate at a lower level in the certificate chain does not need to have all its higher level certificates reverified.

Table D–1 Security Concepts and Definitions

Concept	Definition
Wallet	A wallet is a data structure 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.
Wallet Resource Locator	A wallet resource locator (WRL) provides all the necessary information to locate the wallet. It is a path to an operating system directory which contains a particular wallet.
WRL	See Wallet Resource Locator.
X.509	The public key certificate can be created in various data formats. The X.509 v3 format from ITU is a widely used format.

Starting Oracle Wallet Manager

Refer to your platform-specific documentation for instructions on how to start Oracle Wallet Manager.

Managing Wallets

This section gives you detailed instructions on 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.

This section covers topics in the following subsections:

- Creating a New Wallet
- Opening an Existing Wallet
- Closing a Wallet
- Saving Changes
- Saving a Wallet to a New Location
- Saving in System Default
- Deleting the Wallet
- Changing the Password

Creating a New Wallet

Create a new wallet as follows:

1. Click Wallet > New from the menu bar.

   The New Wallet dialog box is displayed.

2. Read the recommended guidelines for creating a password, then type a password in the Wallet Password field.

3. Retype that password in the Confirm Password field.

4. Click OK to continue.

5. An Alert displays, informs you that a new empty wallet has been created, and prompts you to decide whether you want to create a certificate request: see “Creating a Certificate Request”.

   If you Click Cancel, you are returned to the Oracle Wallet Manager main window. The new wallet you just created is displayed in its left pane. The certificate has a status of Empty, and the wallet displays its default trusted certificates.

6. Click 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.

Note: 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. If a malicious user guesses the password to a user's wallet, then the malicious user could access all the databases that the user can access.

Oracle Corporation recommends that you choose a password that is not too short, is not easily guessed, and has some complexity (such as including a symbol or numeric character). A reasonably strong password has at least six characters, and contains at least one symbol or number (so that it is not a word that can be found in the dictionary), for example, `gol8fer`. Also, it is prudent security practice for users to change their passwords periodically, such as once a month, or once a quarter.

Opening an Existing Wallet

Open a wallet that already exists in the file system directory as follows:

1. Click Wallet > Open from the menu bar.

   The Select Directory dialog box displays.

2. Navigate to the correct directory location in which the wallet is located.
3. Click to select the directory.
4. Click OK.

   The Open Wallet dialog box displays.

5. Type the wallet password in the Wallet Password field.
6. Click OK.
7. A message at the bottom of the window displays the message "Wallet opened successfully".
8. You are returned to the Oracle Wallet Manager main window. The wallet's certificate and its trusted certificates are displayed in the left pane.

Closing a Wallet

Close an open wallet as follows.

1. Click Wallet > Close to close the open wallet in the currently selected directory.
2. A message at the bottom of the window confirms that the wallet is closed.

Saving Changes

Save your changes to the current open wallet as follows.

1. Click Wallet > Save to save changes to the current open wallet.
2. 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 a Wallet to a New Location

Use the Save As option to save the current open wallet to a new directory location.

1. Click Wallet > Save As.

   The Select Directory dialog box displays.
2. Select the directory location in which to save the wallet.
3. Click OK.

Note: You will get the following message if a wallet already exists in the selected directory: "A wallet already exists in the selected path. Do you want to overwrite it?". Click Yes to overwrite the existing wallet, or click No to save the wallet to another directory.

4. 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 will make the current open wallet the wallet that will be used by SSL.

1. Click Wallet > Save in System Default.
2. A message at the bottom of the window confirms that the wallet was successfully saved in the system default wallet location.

Deleting the Wallet

Delete the current open wallet as follows:

1. Click Wallet > Delete.
2. The Delete Wallet dialog box appears.
3. Review the displayed wallet location to verify you are deleting the correct wallet.
4. Type the wallet password.
5. Click OK.
6. A dialog panel appears to inform you that the wallet was successfully deleted.

> **Note:** Any open wallet in an application memory will remain in memory until the application exits. Therefore, deleting a wallet that is currently in use using Oracle Wallet Manager will not immediately affect system operation.

Changing the Password

A password change becomes effective immediately. The wallet is saved to the currently selected directory encrypted with the new password.

Change the password on the current open wallet as follows:

1. Click Wallet > Change Password.

   The Change Wallet Password dialog box appears.
2. Type the existing wallet password.

3. Type the new password. Remember to follow the password guidelines.
4. Re-type the new password.
5. Click OK.

   A message at the bottom of the window informs you that the password was successfully changed.

Managing Certificates

Oracle Wallet Manager uses two kinds of certificates: user certificates and trusted certificates. This section explains how to manage both kinds of certificate, and does so 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 certificate authority. Several trusted certificates are installed by default when you create a new wallet.

Managing User Certificates

Managing user certificates involves performing 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 can not edit an existing certificate request, so only store a correctly filled out certificate request in a wallet.

Create a PKCS #10 certificate request as follows:

1. Click Operations > Create Certificate Request.

   The Create Certificate Request dialog box displays.

2. Type the following information:

Table D–2 Create a Certificate Request Fields and Definitions

Field Name	Description
Common Name	Mandatory—Type the name of the user's or service's identity. Type a user's name in First name Last name format.
Organizational Unit	Optional—Type the name of the identity's organizational unit: for example, Finance.
Organization	Optional—Type the name of the identity's organization, for example, XYZ Corp.
Locality / City	Optional—Type the name of the locality or city in which the identity resides.
State / Province	Optional—Type the full name of the state or province in which the identity resides. Recommendation—Type the full state name, because some certificate authorities do not accept two–letter abbreviations.
Country	Mandatory—Click the drop down list to view a list of country abbreviations. Click to select the country in which the organization is located.
Key Size	Click the drop down box to view a list of key sizes to use when creating the public / private key pair.
Advanced	Click 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. Click OK. An Oracle Wallet Manager dialog box informs you that a certificate request was successfully created. You can now either copy the certificate request text from the body of this dialog panel and paste it into an e-mail message that you send to a certificate authority, or you can export the certificate request to a file.

4. Click 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

You save the certificate request in a file system directory when you elect to export a certificate request.

1. Click Operations > Export Certificate Request from the menu bar.

   The Export Certificate Request dialog box appears.

2. Type the file system directory in which to save your certificate request, or navigate the directory structure under Folders.
3. Type the name of the file to which you want to save your certificate request in the Enter File Name field.
4. Click 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 or file you receive from the certificate authority. Include the lines Begin Certificate and End Certificate.
2. Click Operations > Import User Certificate from the menu bar.

   The Import Certificate dialog box appears.

3. Click the Paste the Certificate radio button, and click 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 click 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. Click Operations > Import User Certificate from the menu bar.
2. Click the Select a file... certificate radio button, and click OK.

   The Import Certificate dialog box appears.
3. Type the path or folder name of the certificate location.
4. Click to select the name of the certificate file (for example, `cert.txt`).
5. Click OK. A message at the bottom of the window displays 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. Click 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. Click Yes. You are returned to the Oracle Wallet Manager main panel, and the certificate will display a status of Requested.

Managing Trusted Certificates

A trusted certificate is the certificate of the issuer of the certificate you requested. Managing trusted certificates consists of performing the tasks discussed in the following sections:

- 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 when you create a new wallet.

Pasting the Trusted Certificate To paste the trusted certificate:

1. Click Operations > Import Trusted Certificate from the menu bar. The Import Trusted Certificate dialog panel appears.
2. Click the Paste the Certificate radio button, and click 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 you received that also contained the user certificate. Include the lines Begin Certificate and End Certificate.
4. Paste the certificate into the window, and click OK. A message at the bottom of the window informs you that the trusted certificate was successfully installed.
5. Click OK.

   You are returned to the Oracle Wallet Manager main panel, and the trusted certificate is displayed at the bottom of the Trusted Certificates tree.

Selecting a File that Contains the Trusted Certificate To select the file:

1. Click Operations > Import Trusted Certificate from the menu bar. The Import Trusted Certificate dialog panel displays.
2. Type the path or folder name of the trusted certificate location.
3. Select the name of the trusted certificate file, for example, `cert.txt`.
4. Click OK. A message at the bottom of the window displays to inform you that the trusted certificate was successfully imported into the wallet.
5. Click OK to dismiss the dialog panel. You are returned to the Oracle Wallet Manager main panel, and the trusted certificate is displayed at the bottom of the Trusted Certificates tree.

Removing a Trusted Certificate

Remove a trusted certificate from a wallet as follows:

1. Select the trusted certificate listed in the Trusted Certificates tree.
2. Click Operations > Remove Trusted Certificate from the menu bar.

   A dialog panel displays and 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. Click Yes. The selected trusted certificate is removed from the Trusted Certificates tree.

Warning: Certificates that are signed by a trusted certificate are no longer verifiable when you remove that trusted certificate 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

Export a trusted certificate to another file system location as follows:

1. Click Operations > Export Trusted Certificate.

   The Export Trusted Certificate dialog box appears.
2. Type the file system directory in which to save your trusted certificate, or click Browse to display the directory structure.
3. Type the name of the file to which you want to save your trusted certificate.
4. Click OK. You are returned to the Oracle Wallet Manager main window.

Exporting All Trusted Certificates

Export all of your trusted certificates to another file system location as follows:

1. Click Operations > Export All Trusted Certificates. The Export Trusted Certificate dialog box appears.
2. Type the file system directory in which to save your trusted certificates, or click Browse to display the directory structure.

3. Type the name of the file to which you want to save your trusted certificates.
4. Click OK. You are returned to the Oracle Wallet Manager main window

Exporting a Wallet

Export a wallet to text-based PKI formats. Individual components will be formatted according to the following standards:

Component	Encoding Standard
Certificate chains	X509v3
Trusted certificates	X509v3
Private keys	PKCS5

E

Using Access Control Directive Format

This appendix describes the format (syntax) of any **Access Control Information Item (ACI)**.

This appendix covers topics in the following sections:

- Schema for orclACI
- Schema for orclEntryLevelACI

Schema for orclACI

The access control directive defined by the user attribute orclACI has the following schema:

```
OrclACI:
{ <oid> 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 = Anonymous
               | 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":
{ <oid> NAME 'orclEntryLevelACI' DESC 'Stores entry level ACL Directive'
EQUALITY accessDirectiveMatch SYNTAX 'orclEntryLevelACIDescription'
USAGE 'directoryOperation' }


<orclEntryLevelACIDescription>
::= access to <object> [by <subject> ( <accessList> )]+
```

F

Oracle Internet Directory Schema Elements

This appendix briefly lists different schema elements supported in the Oracle Internet Directory. Most of these elements are used as defined by the ldapext and ASID working groups of the Internet Engineering Task Force.

> **See Also:** The following URLs on the World Wide Web:
>
> - http://www.ietf.org (for IETF)
> - http://www.ietf.org/html.charters/ldapext-charter.html (ldapext charter and ldap drafts)
> - http://ietf.org/html.charters/asid-charter.html (ASID charter and ldap drafts)
> - http://www.ietf.org/html.charters/ldup-charter.html (for ldup charter and drafts)

This appendix covers topics in the following sections:

- 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 2256	A Summary of the X.500(96) User Schema for use with LDAPv3
URL:	http://www.ietf.org/rfc/rfc2256.txt
RFC 2079	Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs)
URL:	http://www.ietf.org/rfc/rfc2079.txt
RFC 2247	Using Domains in LDAP/X.500 Distinguished Names
URL:	http://www.ietf.org/rfc/rfc2247.txt
RFC 2252	Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions
URL:	http://www.ietf.org/rfc/rfc2252.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/internet-drafts/draft-smith-ldap-inetorgperson-03.txt
Draft:	"Referrals and Knowledge References in LDAP Directories"
URL:	http://www.ietf.org/internet-drafts/draft-ietf-ldapext-referral-00.txt

Proprietary Oracle Internet Directory Schema Elements

Oracle Internet Directory's proprietary schema includes attributes and object classes in the following 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	orclDebugLevel, 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

Table F–1 lists and describes the entire set of configuration set entry attributes that are used to configure an instance of a directory server.

Table F–1 Configuration Set Entry Attributes

Parameter	Description
orcldebuglevel	Debug level associated with this instance of the server. The default for configset0 is 0. The range is 0 to 65535. **See Also:** "Setting Debug Logging Levels by Using the OID Control Utility" on page 5-21 for information on debug levels.
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.
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.
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)

Table F–1 Configuration Set Entry Attributes

Parameter	Description

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/my_wallet`
	On Windows NT, you could set this parameter as follows:
	`file:C:\my_dir\my_wallet`
	For information on setting the location of the Oracle Wallet and the Oracle Wallet password, see Appendix D.

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.

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
- 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

LDAP Schema Description

LDAP Syntax Description

Mail Preference

Master And Shadow Access Points

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

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 above list, Oracle Internet Directory actually enforces the following when it compares attribute values:

DistinguishedNameMatch

caseExactMatch

caseIgnoreMatch

numericStringMatch

IntegerMatch

telephoneNumberMatch

Glossary

Access Control Information 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

The group of access directives that you define. The directives grant levels of access to specific data for specific clients and/or groups of clients.

ACI

See **Access Control Information Item (ACI)**

ACL

See **Access Control List**.

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)**.

ACP

See **Access Control Policy Point**.

Advanced Symmetric Replication (ASR)

A store-and-forward transport feature available in Oracle8*i*. It allows database tables to be kept synchronized across two Oracle databases.

Application Program Interface (API)

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.

administrative area

A subtree on a directory server whose entries are under the control (schema, ACL, and collective attributes) of a single administrative authority.

API

See **Application Program Interface (API)**.

ASR

See **Advanced Symmetric Replication (ASR)**.

attribute

A piece 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.

binding

The process of authenticating to a directory.

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.

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.

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.

consumer

A directory server that is the destination of replication updates.

context prefix

The DN of the root of a **directory naming context**.

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).

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 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.

Directory Replication Group (DRG)

The directory servers participating in a replication agreement.

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.

DIT

See **Directory Information Tree (DIT)**.

DN

See **distinguished name (DN)**.

DRG

See **Directory Replication Group (DRG)**.

DSA

See **Directory System Agent (DSA)**.

DSE

DSA Specific Entries. Different DSAs may hold the same DIT name, but with 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.

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.

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 is different from the GUID for that DN on node2.

GUID

See **global unique identifier (GUID)**.

inherit

When an object class has been derived from another class, it also derives, or *inherits*, many of the characteristics of that other class.

instance

See **server instance**.

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.

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

LDAP

Lightweight Directory Access Protocol. 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.

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.

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

See **directory naming context**.

Net8

The foundation of the Oracle family of networking products, allowing services and their applications to reside on different computers and communicate as peer applications. The main function of Net8 is to establish network sessions and transfer data between a client machine and a server or between two servers. Net8 is located on each machine in the network. Once a network session is established, Net8 acts as a data courier for the client and the server.

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 in the same object class share the same attributes.

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 Monitor

The Oracle Internet Directory component that initiates, monitors, and terminates the Oracle Internet Directory server processes. It also controls the replication server if one is installed.

Oracle Wallet Manager

A Java-based application that security administrators use to manage public-key security credentials on clients and servers.

partition

A unique, non-overlapping directory naming context that is stored on one directory server.

public-key encryption

The process where the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the message is decrypted by the recipient using 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**.

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 **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)**.

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.

schema

The collection of attributes, object classes, and their corresponding matching rules.

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 server instances.

SGA

See **System Global Area (SGA)**.

slave

See **consumer**.

SLAPD

Standalone LDAP daemon.

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. Or it can be viewed as if for each specific aspect of administration, the AAA is partitioned into one or more specific administrative areas.

sponsor node

In replication, the node that is used to provide initial data to a new node.

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

Contains 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.

subACLSubentry

A specific type of subentry that contains ACL information.

subSchemaSubentry

A specific type of **subentry** containing schema 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.

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's 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.

throughput

The overall rate at which directory operations are being completed by Oracle Internet Directory. This is typically represented as “operations per second”.

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.

Index