Oracle® Internet Directory

Administrator's Guide

Release 9.0.2

February 2002

Part No. A95192-01

ORACLE

Oracle Internet Directory Administrator's Guide, Release 9.0.2

Part No. A95192-01

Contributing Authors: Jeffrey Levinger, Sheryl Edwards, Richard Smith

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

Graphic Artist: Valarie Moore

Contents

2 Concepts and Architecture

3 Preliminary Tasks and Information

4 Directory Administration Tools

6 Directory Schema Administration

7 Managing Directory Entries

8 Globalization Support in the Directory

9 The Delegated Administration Service

10 Attribute Uniqueness

Part III Directory Security

11 Directory Security Concepts

12 Secure Sockets Layer (SSL) and the Directory

13 Directory Access Control

15 Oracle Components and Oracle Internet Directory

16 Directory-Based Application Security

17 Directory Storage of User Authentication Credentials

21 High Availability And Failover Considerations

Part V Directory Replication

22 Directory Replication Concepts

23 Oracle Directory Replication Server Administration

24 Addition of a Node by Using the Database Copy Procedure

Part VI The Directory and Clusters

25 Failover in Cluster Configurations

26 Directory Failover in an Oracle9*i* Real Application Clusters Environment

Part VII Directory Plug-ins

27 Oracle Internet Directory Plug-in Framework

30 Oracle Directory Integration Server Administration

31 Security in the Oracle Directory Integration Platform

32 Bootstrapping of a Directory in the Oracle Directory Integration Platform

33 Synchronization with Oracle Human Resources

34 Synchronization with iPlanet Directory Server

35 Synchronization with Third-Party Metadirectory Solutions

36 The Oracle Directory Provisioning Integration Service

B The Access Control Directive Format

C Schema Elements

D Oracle Wallet Manager

E Upgrading Oracle Internet Directory

F Migrating Data from Other LDAP-Compliant Directories

G The LDAP Filter Definition

H Troubleshooting

I Migrating User Data from Application-Specific Repositories

Glossary

Index

Send Us Your Comments

Oracle Internet Directory Administrator's Guide, Release 9.0.2

Part No. A95192-01

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

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

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

- Electronic mail: infodev_us@oracle.com
- FAX: (650) 506-7227 Attn: Server Technologies Documentation Manager
- Postal service:
  Oracle Corporation
  Server Technologies Documentation
  500 Oracle Parkway, Mailstop 4op11
  Redwood Shores, CA 94065
  USA

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

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

Preface

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

This preface contains these topics:

Audience

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

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

Organization

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

Depending on your administrative role, you may find some parts of this guide more pertinent to the tasks you perform.

Table 0–1 Pertinent Sections for Administrative Task Areas

Administrative Task Area	Pertinent Sections of This Guide
Routine administration	Part II: Basic Directory Administration Part III Directory Security
Directory planning and deployment in enterprises and hosted environments	Part III Directory Security Part IV Directory Deployment Part V Directory Replication Part VI: Oracle Internet Directory and Clusters Part VII: Oracle Internet Directory Plug-ins
Integration between Oracle Internet Directory and other directories	Part VIII: The Oracle Directory Integration Platform

Part I: Getting Started

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

Chapter 1, "Introduction"

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

Chapter 2, "Concepts and Architecture"

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

Chapter 3, "Preliminary Tasks and Information"

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

Chapter 4, "Directory Administration Tools"

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

Part II: Basic Directory Administration

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

Chapter 5, "Oracle Directory Server Administration"

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

Chapter 6, "Directory Schema Administration"

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

Chapter 7, "Managing Directory Entries"

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

Chapter 8, "Globalization Support in the Directory"

This chapter discusses Globalization Support as used by Oracle Internet Directory.

Chapter 9, "The Delegated Administration Service"

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

Chapter 10, "Attribute Uniqueness"

This chapter explains the attribute uniqueness feature that enables applications synchronizing with Oracle Internet Directory to use attributes other than distinguished names as their unique keys.

Part III Directory Security

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

Chapter 11, "Directory Security Concepts"

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

Chapter 12, "Secure Sockets Layer (SSL) and the Directory"

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

Chapter 13, "Directory Access Control"

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

Part IV Directory Deployment

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

Chapter 14, "General Deployment Considerations"

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

Chapter 15, "Oracle Components and Oracle Internet Directory"

Many Oracle components use Oracle Internet Directory for a variety of purposes. In doing this, they rely on a consolidated Oracle Internet Directory schema and a default Directory Information Tree (DIT). This chapter:

- Describes the consolidated Oracle Internet Directory schema used by various components
- Describes a default DIT structure available when using the various Oracle components

Chapter 16, "Directory-Based Application Security"

This chapter discusses how you can exploit the way Oracle Internet Directory stores access control policies to secure applications in a large enterprise and in hosted environments.

Chapter 17, "Directory Storage of User Authentication Credentials"

This chapter explains how Oracle components store application security credentials in Oracle Internet Directory to make their administration easy for both end users and administrators and to address a major security threat to any enterprise.

Chapter 18, "Password Policies"

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

Chapter 19, "Capacity Planning Considerations"

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

Chapter 20, "Tuning Considerations"

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

Chapter 21, "High Availability And Failover Considerations"

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

Part V Directory Replication

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

Chapter 22, "Directory Replication Concepts"

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

Chapter 23, "Oracle Directory Replication Server Administration"

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

Chapter 24, "Addition of a Node by Using the Database Copy Procedure"

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

Part VI: Oracle Internet Directory and Clusters

Part VI discusses cluster support in Oracle Internet Directory.

Chapter 25, "Failover in Cluster Configurations"

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

Chapter 26, "Directory Failover in an Oracle9i Real Application Clusters Environment"

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

Part VII: Oracle Internet Directory Plug-ins

Chapter 27, "Oracle Internet Directory Plug-in Framework"

This chapter describes how you can extend the capabilities of the Oracle directory server by using plug-ins developed by either Oracle Corporation or third-party vendors.

Part VIII: The Oracle Directory Integration Platform

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

Chapter 28, "Oracle Directory Integration Platform Concepts and Components"

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

Chapter 29, "The Oracle Directory Synchronization Service"

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

Chapter 30, "Oracle Directory Integration Server Administration"

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

Chapter 31, "Security in the Oracle Directory Integration Platform"

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

Appendix C, "Schema Elements"

This appendix lists schema elements supported in Oracle Internet Directory.

Appendix D, "Oracle Wallet Manager"

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

Appendix E, "Upgrading Oracle Internet Directory"

This appendix tells you how to upgrade to Oracle Internet Directory Release 9.0.2 from Oracle Internet Directory release 2.1.1.

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

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

Appendix G, "The LDAP Filter Definition"

This appendix, copied with permission from the **Internet Engineering Task Force (IETF)**, describes a directory access protocol that provides both read and update access.

Appendix H, "Troubleshooting"

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

Appendix I, "Migrating User Data from Application-Specific Repositories"

This appendix explains how to migrate data from application-specific repositories by first creating an intermediate template file, and then running the OID Migration Tool.

Related Documentation

For more information, see:

- Online help available through Oracle Directory Manager, the Delegated Administration Service, and Oracle Enterprise Manager
- The Oracle9*i* Application Server and Oracle9*i* Database Server documentation sets, especially:
  - *Oracle Internet Directory Application Developer's Guide*
  - *Oracle9i Database Administrator's Guide*

- *Oracle9i Application Developer's Guide - Fundamentals*
- *Oracle9i Application Server Administrator's Guide*
- *Oracle9i Net Services Administrator's Guide*
- *Oracle9i Real Application Clusters Administration*
- *Oracle9i Replication*
- *Oracle Advanced Security Administrator's Guide*

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

`http://oraclestore.oracle.com/`

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

`http://www.oraclebookshop.com/`

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

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

`http://otn.oracle.com/admin/account/membership.html`

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

`http://otn.oracle.com/docs/index.htm`

To access the database documentation search engine directly, please visit

`http://tahiti.oracle.com`

For additional information, see:

- Chadwick, David. *Understanding X.500—The Directory.* Thomson Computer Press, 1996.
- Howes, Tim and Mark Smith. *LDAP: Programming Directory-enabled Applications with Lightweight Directory Access Protocol.* Macmillan Technical Publishing, 1997.
- Howes, Tim, Mark Smith and Gordon Good, *Understanding and Deploying LDAP Directory Services.* Macmillan Technical Publishing, 1999.

- Internet Assigned Numbers Authority home page, `http://www.iana.org`, for information about object identifiers
- Internet Engineering Task Force (IETF) documentation, especially:
  - `http://www.ietf.org` for the IETF home page
  - `http://www.ietf.org/html.charters/ldapext-charter.html` for the ldapext charter and LDAP drafts)
  - `http://www.ietf.org/html.charters/ldup-charter.html` for the LDUP charter and drafts
  - `http://www.ietf.org/rfc/rfc2254.txt`, "The String Representation of LDAP Search Filters"
  - `http://www.ietf.org/rfc/rfc1823.txt`, "The LDAP Application Program Interface"
- The OpenLDAP Community, `http://www.openldap.org`

Conventions

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

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

Conventions in Text

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

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle9i Database Concepts* Ensure that the recovery catalog and target database do *not* reside on the same disk.

Convention	Meaning	Example
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. **Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to open SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepUtil class implements these methods.
lowercase italic monospace (fixed-width) font	Lowercase italic monospace font represents placeholders or variables.	You can specify the *parallel_clause*. Run U*old_release*.SQL where *old_release* refers to the release you installed prior to upgrading.

Conventions in Code Examples

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

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

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

Convention	Meaning	Example
[ ]	Brackets enclose one or more optional items. Do not enter the brackets.	`DECIMAL (digits [ , precision ])`
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	`{ENABLE | DISABLE}`
|	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	`{ENABLE | DISABLE}` `[COMPRESS | NOCOMPRESS]`

...	Horizontal ellipsis points indicate either:
	■ That we have omitted parts of the code that are not directly related to the example
	■ That you can repeat a portion of the code
		`CREATE TABLE ... AS subquery;`
		
		`SELECT col1, col2, ... , coln FROM employees;`

. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	`SQL> SELECT NAME FROM V$DATAFILE;`
		`NAME`
		`----------`
		`/fsl/dbs/tbs_01.dbf`
		`/fs1/dbs/tbs_02.dbf`
		`.`
		`.`
		`.`
		`/fsl/dbs/tbs_09.dbf`
		`9 rows selected.`

Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	`acctbal NUMBER(11,2);` `acct CONSTANT NUMBER(4) := 3;`
Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	`CONNECT SYSTEM/system_password` `DB_NAME = database_name`

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

Convention	Meaning	Example

lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.
	
	Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.
		`SELECT last_name, employee_id FROM employees;`
		`sqlplus hr/hr`
		`CREATE USER mjones IDENTIFIED BY ty3MU9;`

Conventions for Microsoft Windows Operating Systems

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

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (|), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	`c:\winnt"\"system32 is the same as C:\WINNT\SYSTEM32`
`C:\>`	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the *command prompt* in this manual.	`C:\oracle\oradata>`

Convention	Meaning	Example
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	`C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\"` `C:\>imp SYSTEM/`*`password`*` FROMUSER=scott TABLES=(emp, dept)`
HOME_NAME	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	`C:\> net start Oracle`*`HOME_NAME`*`TNSListener`
ORACLE_HOME and *ORACLE_BASE*	In releases prior to Oracle8*i* release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory that by default used one of the following names: ▪ `C:\orant` for Windows NT ▪ `C:\orawin98` for Windows 98 This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle`. If you install Oracle9*i* release 1 (9.0.1) on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is `C:\oracle\ora90`. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. Refer to *Oracle9i Database Getting Started for Windows* for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.	Go to the *ORACLE_BASE*`\`*`ORACLE_HOME`*`\rdbms\admin` directory.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

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

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

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

What’s New in Oracle Internet Directory?

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

New Features Introduced with Oracle Internet Directory Release 9.0.2

This section describes the new features introduced with Oracle Internet Directory Release 9.0.2.

- **Server-side entry caching**—This feature reduces directory query latency for LDAP clients. By configuring a server-side entry cache based on naming context, identity of client, or other available parameters, Oracle Internet Directory ensures that previously retrieved entries and their attributes are stored in shared memory, and are thus available to subsequent data requestors. Queries that conform to the configured parameters then need only retrieve a small subset of data—internal globally unique identifiers (GUIDs)—for filter-matching entries from the directory. These returned GUIDs are then used as a fast lookup mechanism into the cached entry and attribute data, which is then returned to the client.

  See Also: "Entry Caching" on page 20-12

- **New directory integration capabilities**—Oracle Internet Directory Release 9.0.2 introduces new kinds of connectivity with other applications and repositories, both Oracle-built and otherwise. The new Oracle Directory Provisioning Integration Service and Oracle Directory Synchronization Service are built upon the Oracle Directory Integration Platform (introduced with Oracle Internet Directory v2.1.1.1 in the Oracle8*i* Release 3 timeframe).

  - **Oracle Directory Provisioning Integration Service**—Provisioning is the process of granting or revoking a user's access to application resources based on business rules. The user may be either a human end user or an application.

    The Oracle Directory Provisioning Integration Service ensures that subscribing applications or business entities are alerted to updates in Oracle Internet Directory for the purpose of keeping local repositories synchronized. It enables you to synchronize local, application-specific information by using Oracle Internet Directory as a source of truth.

  - **Oracle Directory Synchronization Service and the LDAP connector**—The Oracle Directory Synchronization Service enables near-complete leveraging of previously-deployed infrastructure, including but not limited to ERP and CRM systems, third-party LDAP directories, and NOS user repositories. It enables you to synchronize information between enterprise directories and

Oracle Internet Directory. This allows for centralized administration, thereby reducing administrative costs. It ensures that data is consistent and up-to-date across the enterprise.

See Also: Chapter 28, "Oracle Directory Integration Platform Concepts and Components"

- **Enterprise password policy management enhancements**—You can now construct password policies to ensure:
  - Expiration dates
  - Grace periods
  - Minimum password lengths
  - Approved password syntaxes and retry limits
  - Lockout of those attempting to gain illicit access to the directory service after a certain number of failed attempts

  You can now use salted SHA as a hashing algorithm. This means that you can now select from these available hashing algorithms:

  - **MD4**—A one-way hash function that produces a 128-bit hash
  - **MD5**—An improved, and more complex, version of MD4
  - **SHA**—Secure Hash Algorithm, which produces a 160-bit hash, longer than MD5. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.

    You can also use salted SHA. A salt is a random number added to and stored with the hash value. It prevents pre-computed dictionary attacks by making it extremely expensive to recover the value that was originally hashed.
  - **UNIX Crypt**—The UNIX encryption algorithm
  - No Hashing

See Also:

- "Protection of User Passwords for Directory Authentication" on page 11-7 for a conceptual discussion
- Chapter 18, "Password Policies" for instructions on setting password hashing

- **Attribute uniqueness—**In the prior Oracle Internet Directory architecture, the only way to enforce attribute uniqueness was to make an attribute a part of your DN. This worked well with the user identifier (if used as the RDN), but it was not always appropriate and easy to configure. Within a level of a branch of the tree, it was guaranteed to be unique. For example, if your DN was `uid=dlin, ou=people, o=oracle`, then this would be unique directly under `ou=people`. However, you could have the same user identifier in another branch—for example, `uid=dlin, ou=others, o=oracle`. In short, attribute uniqueness was guaranteed only under a given branch, and only within one level.

  The applications Oracle Internet Directory synchronizes with can use attributes other than DN as their unique keys. The ability of Oracle Internet Directory to enforce attribute uniqueness enables all applications their own notions of "user," to synchronize their user base with a user repository stored in an enterprise's Oracle Internet Directory server.

- **Multiple password verifier support**—Oracle Internet Directory can now store passwords for multiple applications and protocols. For example, four-digit Personal Identification Numbers (PINs) for voicemail can sit alongside longer alphanumeric single sign-on passwords and X509 v3 digital certificates for the same user. This new feature gives the application developer far greater flexibility for directory-enabling their product stack.

  See Also: Chapter 17, "Directory Storage of User Authentication Credentials"

- **Expanded proxy user capabilities**—This new feature enables a developer to exploit the power of the middle tier more effectively. Users no longer need to establish independent, unrelated sessions with the directory. If a middle-tier fromOracle9*i* Application Server or elsewhere invokes the proxy user bind method on behalf of numerous clients in succession, then Oracle Internet

Directory respects each client's credential and privileges respectively, even though the agent doing the actual binding remains unchanged throughout.

See Also:

- Chapter 11, "Directory Security Concepts"
- "Managing Super Users, Guest Users, and Proxy Users" on page 5-18

- **Integration with Oracle9*i* Application Server components**—Through the Oracle Directory Provisioning Integration Service, Oracle Internet Directory Release 9.0.2 serves as a central component of the Oracle9*i* Application Server. Every component of Oracle9*i* Application Server now uses Oracle Internet Directory for storing common cross-component metadata, such as valid user identifiers and their passwords.

  See Also: Chapter 15, "Oracle Components and Oracle Internet Directory"

- **Delegated Administration Service Self-Service Console**—This new feature enables you to flexibly administer applications, subscribers, and end users either from a central team or through decentralization and delegation.

  The Delegated Administration Service Self-Service Console gives authorized end users a view of their personalized preferences and the ability to update their Oracle9*i*AS Single Sign-On password. It provides an intuitive user interface for searching for people and other directory-based resource information within Oracle Internet Directory.

  You can use the Self-Service Console to configure the object classes, user groups, permissions, and other elements of directory information metadata stored in Oracle Internet Directory.

  See Also: Chapter 9, "The Delegated Administration Service"

- **Enterprise Manager integration**—You can start, stop, and monitor Oracle Internet Directory instances by using the standard, newly-enhanced Enterprise Manager console. You can perform system diagnostics on running Oracle

Internet Directory instances, and generate performance graphs to determine ongoing performance and peak load times.

See Also: Monitoring, Debugging, and Auditing the Directory Server on page 5-22

- **Oracle Directory Manager enhancements**—Oracle Internet Directory's standalone, 100% Java administration console, Oracle Directory Manager, has now evolved in many ways. You can use it to:
  - Configure hosted subscriber domains
  - Construct password policies
  - Configure Oracle Directory Synchronization Service and Oracle Internet Directory connectors and agents

  In general, any directory-specific configuration or maintenance task not available at the high-level OEM GUI is now doable through ODM, as well as command-line interfaces supplied with Oracle Internet Directory.

  See Also: Chapter 4, "Directory Administration Tools"

- **Server-side plug-in framework**—This new feature enables directory applications to roll out advanced capabilities such as referential integrity / cascading deletions of LDAP objects, external authentication of directory clients, brokered access, and synchronization with external relational tables. The plug-ins are executable before or after an LDAP command takes place, without the traditional risks of such technologies.

  See Also: Chapter 27, "Oracle Internet Directory Plug-in Framework"

- **Entry alias dereferencing**—The LDAP v3 standard requires that all entries in a directory have globally unique identifiers known as distinguished names. These are typically fairly long and cumbersome to use, so Oracle Internet Directory provides this new feature to automatically dereference IETF-standard alias objects used to point to a fully-qualified LDAP distinguished name. For example, "DavesServer1" can be used as an entry alias or pointer to the actual

directory entry named `dc=server1, dc=us, dc=oracle, dc=com`. Oracle Internet Directory stores, parses, and chases all alias references for complete client-side transparency.

See Also: "Dereferencing Alias Entries" on page 5-37

- **Delegated Administration Service enhancements**

  Administrators can now use the Delegated Administration Service and its accompanying console to:

  - Create other regional or departmental administrators
  - Grant them specific, delegated permissions to administer users for a particular region or department

  The Oracle Internet Directory Self-Service Console provides a unified resource for directory administrators, directory service subscribers, and end users.

See Also: Chapter 9, "The Delegated Administration Service"

- **Upgrade procedures**

  These procedures enable you to upgrade from Oracle Internet Directory release 2.1.1. and release 3.0.1.

See Also: Appendix E, "Upgrading Oracle Internet Directory"

New Features Introduced with Oracle Internet Directory Release 3.0.1

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

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

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

See Also: "Running Multiple Installations of Oracle Internet Directory on One Host" on page 14-12

- **Delegated Administration Service**

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

  See Also: Chapter 9, "The Delegated Administration Service"

- **Failover in cluster configurations**

  This new feature enables you to increase high availability by using logical hosts—as opposed to physical hosts—in clustered environments.

  See Also: Chapter 25, "Failover in Cluster Configurations"

- **Failover in an Oracle Real Application Clusters environment**

  Oracle9*i* Real Application Clusters is a computing environment that harnesses the processing power of multiple, interconnected computers. Along with a collection of hardware, called a cluster, it unites the processing power of each component to become a single, robust computing environment. A cluster comprises two or more computers, also called nodes.

  You can run Oracle Internet Directory in an Oracle Real Application Clusters system.

  See Also: Chapter 26, "Directory Failover in an Oracle9i Real Application Clusters Environment"

- **Support for logical hosts**—Oracle Internet Directory Release 3.0.1 enables you to increase high availability by using *logical hosts* – as opposed to physical hosts – in clustered environments. A logical host consists of one or more disk groups, and pairs of host names and IP addresses. It is mapped to a physical host in the cluster. This physical host services the host name and IP address of the logical host.

In this paradigm, the directory server binds to the logical host, rather than the physical host. It maintains this connection even if the logical host fails over to a new physical host.

A client connects to the directory server by using the logical host name and address of the server. If the logical host fails over to a new physical host, then that failover is transparent to the client.

- **Oracle Directory Integration Platform**

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

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

- **Password policy management**

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

  See Also:

  - "Password Policies" for a conceptual discussion
  - Chapter 18, "Password Policies"

- **Performance and scalability enhancements**
- **Upgrade procedures**

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

  See Also: Appendix E, "Upgrading Oracle Internet Directory"

- **UTF8 restriction removed**

  The Oracle directory server and database tools are no long restricted to run on a UTF8 database.

New Features Introduced with Oracle Internet Directory Release 2.1.1

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

- **Attribute options, including language codes**

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

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

  See Also:

  - "Attribute Options" on page 2-7 for a conceptual discussion
  - "Managing Entries with Attribute Options by Using Oracle Directory Manager" on page 7-11
  - "Managing Entries with Attribute Options by Using Command-Line Tools" on page 7-15

- **Change log purging enhancements**

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

  See Also:

  - "Change Log Purging" on page 22-6 for a conceptual discussion
  - "Modifying Directory Replication Server Configuration Parameters" on page 23-13

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

- **modifiersName**
- **modifyTimestamp**

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

See Also:

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

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

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

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

- **Object class explosion**

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

  See Also: ""Guidelines for Adding Object Classes" on page 6-3 for an explanation of how to use this feature when adding object classes

- **OID database statistics collection tool**

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

See Also: "Using the OID Database Statistics Collection Tool" on page 4-15

- **Password protection enhancements**

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

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

  See Also:

  - "Protection of User Passwords for Directory Authentication" on page 11-7 for a conceptual discussion
  - Chapter 18, "Password Policies" for instructions on setting password hashing

- **Replication tools**

  The following new replication tools are now added:

  - **Human intervention queue manipulation tool**

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

  - **OID reconciliation tool**

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

See Also:

- "Using the Replication Tools" on page 4-14 for a brief explanation of this tool
- "Using the Human Intervention Queue Manipulation Tool" on page 23-31
- "Using the OID Reconciliation Tool" on page 23-31

- **Replication node deletion**

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

  See Also: "Deleting a Replication Node" on page 23-27

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

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

  Note: This feature was replaced in Release 9.0.2 by the Oracle Directory Integration Platform. See Chapter 28, "Oracle Directory Integration Platform Concepts and Components" for further information.

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

  These new procedures enable you to upgrade from either Oracle Internet Directory release 2.0.4.x or release 2.0.6. Not supported in release 2.1.1.1 or in Release 9.0.2.

  See Also: Appendix E, "Upgrading Oracle Internet Directory"

Part I

Getting Started

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

1

Introduction

This chapter introduces online directories, provides an overview of the Lightweight Directory Application Protocol (LDAP) version 3, and explains some of the unique features and benefits of Oracle Internet Directory.

This chapter contains these topics:

What Is a Directory?

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

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

This section contains these topics:

The Expanding Role of Online Directories

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

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

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

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

Online Directories	Relational Databases
Primarily read-focused. Typical use involves a relatively small number of data updates, and a potentially large number of data retrievals.	**Primarily write-focused.** Typical use involves continuous recording of transactions, with retrievals done relatively infrequently.
Designed to handle relatively simple transactions on relatively small units of data. For example, an application might use a directory simply to store and retrieve an e-mail address, a telephone number, or a digital portrait.	**Designed to handle large and diverse transactions using many operations on large units of data.**
Designed to be location-independent. Directory applications expect, at all times, to see the same information throughout the deployment environment—regardless of which server they are querying. If a queried server does not store the information locally, then it must either retrieve the information or point the client application to it transparently.	**Typically designed to be location-specific.** While a relational database can be distributed, it usually resides on a particular database server.
Designed to store information in entries. These entries might represent any resource customers wish to manage: employees, e-commerce partners, conference rooms, or shared network resources such as printers. Associated with each entry is a number of attributes, each of which may have one or more values assigned. For example, typical attributes for a `person` entry might include first and last names, e-mail addresses, the address of a preferred mail server, passwords or other login credentials, or a digitized portrait.	**Designed to store information as rows in relational tables.**

The Problem: Too Many Special Purpose Directories

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

Managing so many special purpose directories can cause problems:

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

Today's enterprises need a more general purpose directory infrastructure, one based on a common standard for supporting a wide variety of applications and services.

What Is LDAP?

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

This section contains these topics:

LDAP and Simplified Directory Management

LDAP was conceived as an Internet-ready, lightweight implementation of the International Standardization Organization (ISO) X.500 standard for directory

services. It requires a minimal amount of networking software on the client side, which makes it particularly attractive for Internet-based, thin client applications.

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

- It provides all users and applications in the enterprise with a single, well-defined, standard interface to a single, extensible directory service. This makes it easier to rapidly develop and deploy directory-enabled applications.
- It reduces the need to enter and coordinate redundant information in multiple services scattered across the enterprise.
- Its well-defined protocol and array of programmatic interfaces make it more practical to deploy Internet-ready applications that leverage the directory.

LDAP Version 3

The most recent version of LDAP, Version 3, was approved as a proposed Internet Standard by the **Internet Engineering Task Force (IETF)** in December 1997. LDAP Version 3 improves on LDAP Version 2 in several important areas:

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

See Also:

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

What Is Oracle Internet Directory?

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

This section contains these topics:

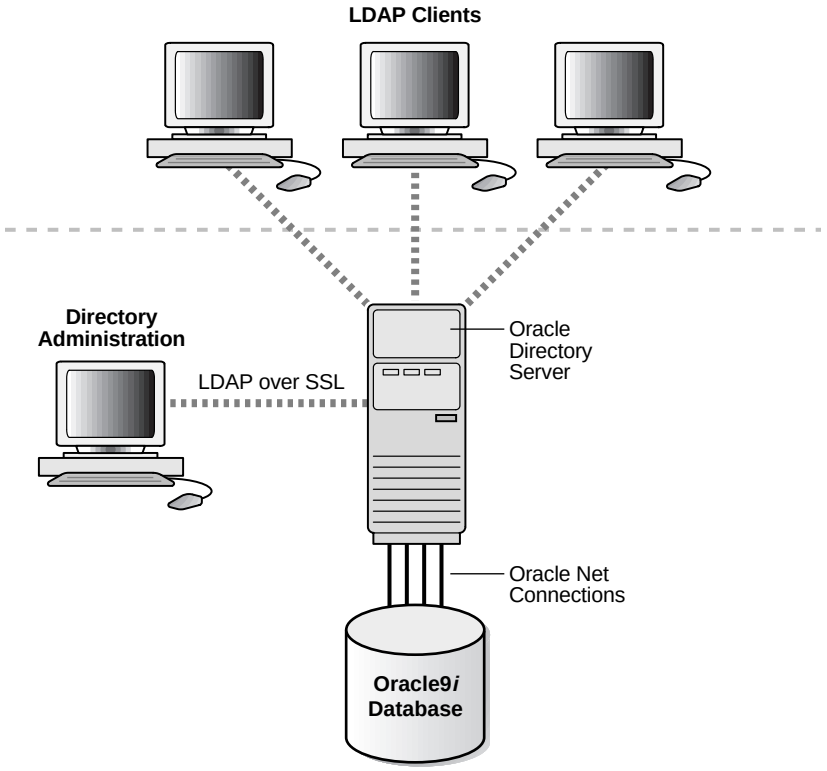
- Architecture of the Oracle Internet Directory
- Components of the Oracle Internet Directory
- Advantages of Oracle Internet Directory

Architecture of the Oracle Internet Directory

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

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

Figure 1–1 Oracle Internet Directory Architecture



Components of the Oracle Internet Directory

Oracle Internet Directory includes:

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

- Directory administration tools, which include:
  - Oracle Directory Manager, which simplifies directory administration through a Java-based graphical user interface
  - A variety of command-line administration and data management tools invoked from LDAP clients
  - The Delegated Administration Service (DAS), which frees global directory administrators for the more important and complex tasks of directory management. DAS does this by providing an easy-to-use web interface with the following advantages:
    * End users can modify their own passwords without the intervention of an administrator.
    * Delegated administrators, such as non-technical managers, can create and manage both users and groups.
    * All users can search parts of the directory to which they have access.
  - Tools within the web-based interface for Oracle Enterprise Manager, for managing OID server instances. These tools enable an administrator to monitor real-time events and statistics from a normal browser and, if desired, to start the process of collecting future such data into a new historical repository.
- Oracle Directory Integration Platform, including Oracle directory integration server, which enables you to synchronize connected directories and subscribed applications with Oracle Internet Directory. You can also use the Oracle Directory Integration Platform to develop and deploy your own connectivity agents.

  See Also: Part VIII: The Oracle Directory Integration Platform for more information about the Oracle Directory Integration Platform

Advantages of Oracle Internet Directory

Among its more significant benefits, Oracle Internet Directory provides scalability, high availability, security, and tight integration with the Oracle environment.

Scalability

Oracle Internet Directory exploits the strengths of Oracle9*i*, enabling support for terabytes of directory information. In addition, such technologies as multithreaded

LDAP servers and database connection pooling allow it to support thousands of concurrent clients with subsecond search response times.

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

High Availability

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

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

Security

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

Integration with the Oracle Environment

All Oracle products use Oracle Internet Directory. Through the Oracle Directory Integration Platform, Oracle Internet Directory provides a single point of integration between the Oracle environment and other directories such as NOS directories, third-party enterprise directories, and application-specific user repositories.

How Oracle Products Use Oracle Internet Directory

Oracle Internet Directory enables Oracle components to achieve easier and more cost-effective administration of the application environment; tighter security

through centralized security policy administration; and a single point of integration between distributed enterprise directories. This section describes a few examples.

Easier and More Cost-Effective Administration

Oracle Net Services uses Oracle Internet Directory to store and resolve database services and the simple names, called net service names, that can be used to represent them. In client connect strings, net service names serve as connect identifiers. The directory server resolves these connect identifiers to connect descriptors, which are passed back to the client.

Oracle Unified Messaging uses Oracle Internet Directory:

- To store and retrieve server configuration information, e-mail-specific user preferences, and users' recorded voice mail greetings
- Validate email recipient lists
- Represent and manage email distribution lists
- To store runtime parameters, thereby enabling Oracle Unified Messaging administrators to easily manage distributed installations

The self-service, integrated enterprise portals that use **Oracle Portal** access Oracle Internet Directory to store common user and group attributes. The Oracle Portal administration tool also leverages the Delegated Administration Service for certain tasks.

Tighter Security Through Centralized Security Policy Administration

Oracle9*i* uses Oracle Internet Directory to store user names and passwords, and it authenticates users by using LDAP mechanisms instead of SSL. It uses Oracle Internet Directory to store a password verifier along with the entry of each user.

Oracle Advanced Security uses Oracle Internet Directory for:

- Central Management of user authentication credentials

  Oracle Advanced Security stores a user's database password in the directory as an attribute of his or her user entry, instead of in each database.

- Central management of user authorizations

  Oracle Advanced Security uses directory entries called enterprise roles to determine what privileges a given enterprise user has within a given schema, shared or owned. Enterprise roles are containers for database-specific global roles. For example, a user might be assigned the enterprise role clerk, which

might contain the global role hrclerk and its attendant privileges on the human resources database and the global role analyst and its attendant privileges on the payroll database.

- Mappings to shared schemas

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

- Single password authentication

  In Oracle9*i*, Oracle Advanced Security enables enterprise users to authenticate to multiple databases by using a single, centrally managed password. The password is stored in the directory as an attribute of the user's entry and is protected by encryption and access control lists. This feature eliminates the overhead associated with setting up Secure Sockets Layer (SSL) on clients and frees users from having to remember multiple passwords.

- Enterprise user security

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

- Central storage of PKI credentials

  In Oracle9*i*, user wallets can be stored in the directory as an attribute of the user's entry. This feature enables mobile users to retrieve and open their wallets by using Enterprise Login Assistant. While the wallet is open, authentication is transparent—that is, users can access any database on which they own or share a schema without having to authenticate again.

Oracle9*i*AS Single Sign-On uses Oracle Internet Directory to store user entries. It maps users for any partner application to user entries in Oracle Internet Directory entries, and authenticates them by using LDAP mechanisms.

Integration of Distributed Directories

The Oracle Directory Integration Platform is a collection of interfaces and services for integrating multiple directories by using Oracle Internet Directory as the central directory.

The Oracle Directory Integration Platform provides these benefits:

- Because all Oracle components are pre-integrated with Oracle Internet Directory, you do not need to integrate each component with a directory service.
- You can integrate the entire Oracle environment with third-party directories simply by integrating each third-party directory with Oracle Internet Directory. You do not need to painstakingly integrate each application with each directory.

2

Concepts and Architecture

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

This chapter contains these topics:

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

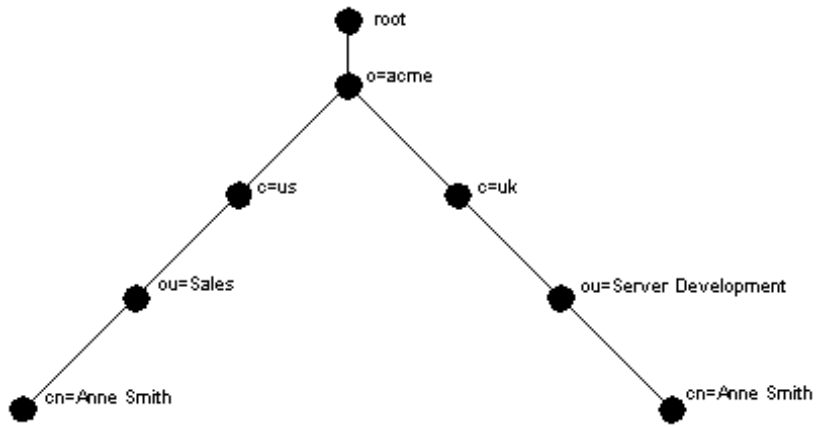
Entries

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

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

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

Figure 2–1 A Directory Information Tree



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

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

The DN for this "Anne Smith" entry is:

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

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

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

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

To make operations on entries quick and efficient, Oracle Internet Directory assigns a unique identifier to each entry, then stores a specified number of those identifiers in cache memory. When a user performs an operation on an entry, the directory server looks in the cache for the entry identifier, then retrieves the corresponding entry from the directory. This method, called entry caching, enhances Oracle Internet Directory performance, and is especially useful in smaller and medium-sized enterprises.

Note: In Oracle Internet Directory Release 9.0.2, you can use entry caching only in the case of a single server, single instance Oracle Internet Directory node.

See Also: Chapter 7, "Managing Directory Entries."

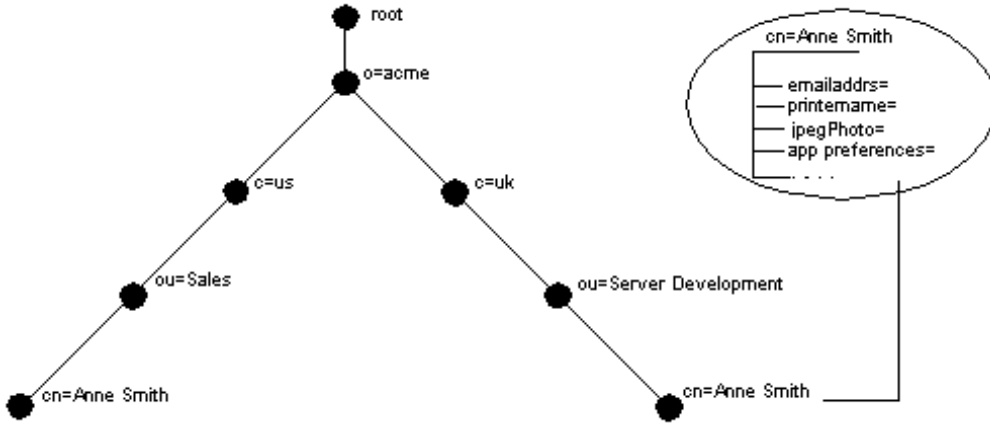
Attributes

In a typical telephone directory, an **entry** for a person contains such information items as an address and a phone number. In an online directory, such an

information item is called an **attribute**. Attributes in a typical employee entry can include, for example, a job title, an e-mail address, or a phone number.

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

Figure 2–2 Attributes of the Entry for Anne Smith



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

This section contains these topics:

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

Kinds of Attribute Information

Attributes contain two kinds of information.

- Application Information

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

- Operational Information

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

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

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

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

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

> **See Also:** "Setting System Operational Attributes" on page 5-13 for instructions on configuring system operational attributes

Single-Valued and Multivalued Attributes

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

Common LDAP Attributes

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

Table 2–1 Common LDAP Attributes

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

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

Attribute Syntax

Attribute syntax is the format of the data that can be loaded into each attribute. For example, the syntax of the `telephoneNumber` attribute might require a telephone number to be a string of numbers containing spaces and hyphens. However, the syntax for another attribute might require specifying whether the data has to be in

the form of a date, or whether the data can consist of numbers only. Each attribute must have one and only one syntax attached to it.

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

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

Attribute Matching Rules

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

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

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

Attribute Options

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

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

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

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

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

See Also:

- "Managing Entries with Attribute Options by Using Oracle Directory Manager" on page 7-11
- "Managing Entries with Attribute Options by Using Command-Line Tools" on page 7-15

Object Classes

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

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

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

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

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

This section contains these topics:

- Subclasses, Superclasses, and Inheritance
- Object Class Types

Subclasses, Superclasses, and Inheritance

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

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

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

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

Object Class Types

There are three types of object classes:

- Abstract
- Structural
- Auxiliary

Abstract Object Classes

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

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

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

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

Structural Object Classes

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

Auxiliary Object Classes

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

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

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

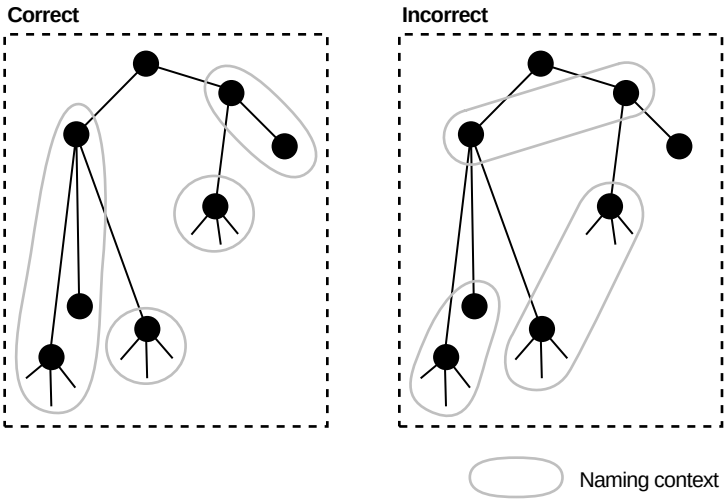
See Also: Chapter 6, "Directory Schema Administration."

Naming Contexts

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

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

Figure 2–3 Valid and Invalid Naming Contexts



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

> **See Also:** "Managing Naming Contexts" on page 5-17 for instructions on how to publish a naming context

The Directory Schema

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

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

See Also:

- Chapter 6, "Directory Schema Administration."
- Appendix C, "Schema Elements" for a list of both standard and proprietary schema elements installed with Oracle Internet Directory

Security

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

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

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

applications in their departments. These department administrators can then control access to their department applications.

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

Globalization Support

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

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

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

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

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

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

See Also:

- "Oracle Internet Directory Architecture" on page 2-15 for information on the main Oracle Internet Directory components
- Chapter 8, "Globalization Support in the Directory" for instructions on using Globalization Support in Oracle Internet Directory
- *Oracle9i Database Globalization Support Guide* in the Oracle Database Documentation Library for a detailed discussion of Globalization Support

---

Note: The Oracle directory server and database tools are no longer restricted to run on a UTF8 database. However, Oracle Corporation recommends that the client and database character sets be the same if the database underlying the Oracle Internet Directory Server is not UTF8. Otherwise, there may be data loss during LDAP add, delete, modify, or modifydn operations if the client data cannot be mapped to the database character set.

Oracle Internet Directory Architecture

This section contains these topics:

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

An Oracle Internet Directory Node

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

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

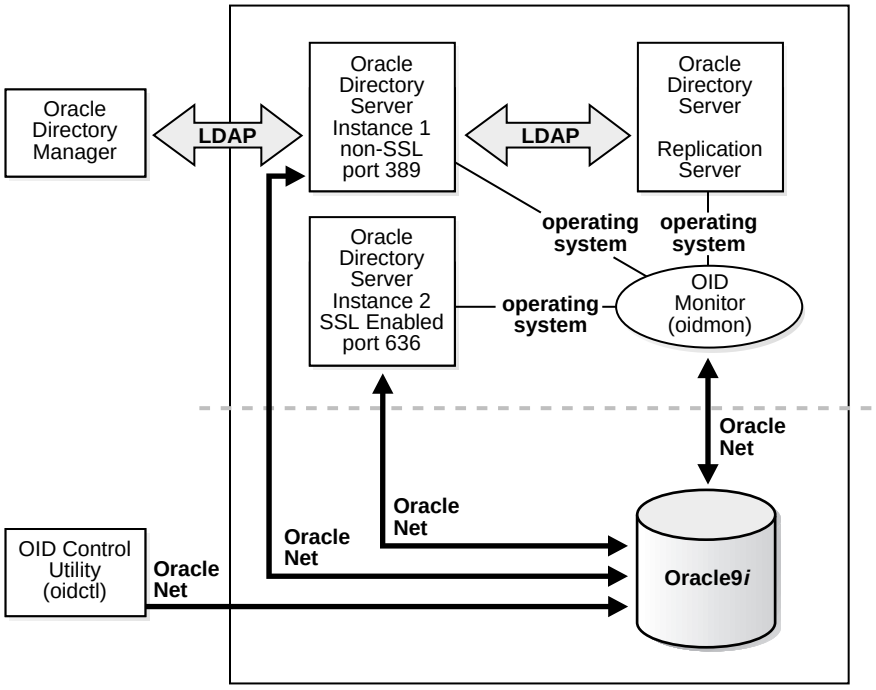
- The **OID Control Utility**
- The Oracle directory server instance 1 non-SSL port 389
- The Oracle directory server instance 2 SSL-enabled port 636
- The **OID Monitor**

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

- Oracle Directory Manager
- Oracle directory replication server

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

Figure 2–4 A Typical Oracle Internet Directory Node



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

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

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

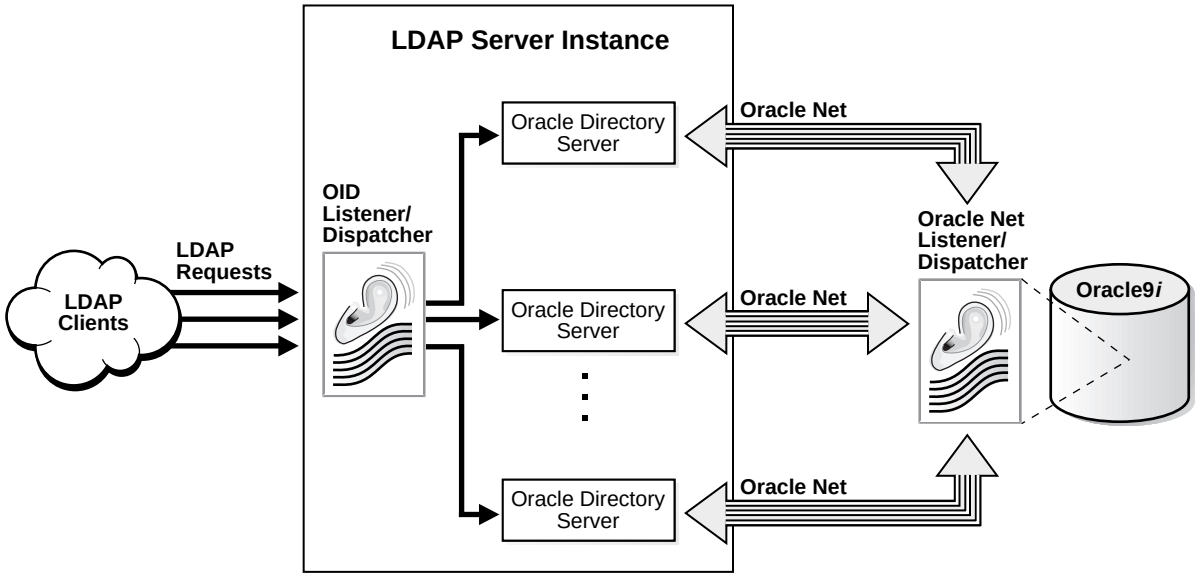
Component	Description
OID Monitor (OIDMON)	Initiates, monitors, and terminates the LDAP server processes. If you elect to install a replication server, OID Monitor controls it. When you issue commands through OID Control Utility (OIDCTL) to start or stop directory server instances, your commands are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

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

An Oracle Directory Server Instance

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

Figure 2–5 Oracle Directory Server Instance Architecture



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

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

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

Configuration Set Entries

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

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

See Also:

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

Example: How Oracle Internet Directory Works

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

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

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

Distributed Directories

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

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

This section contains these topics:

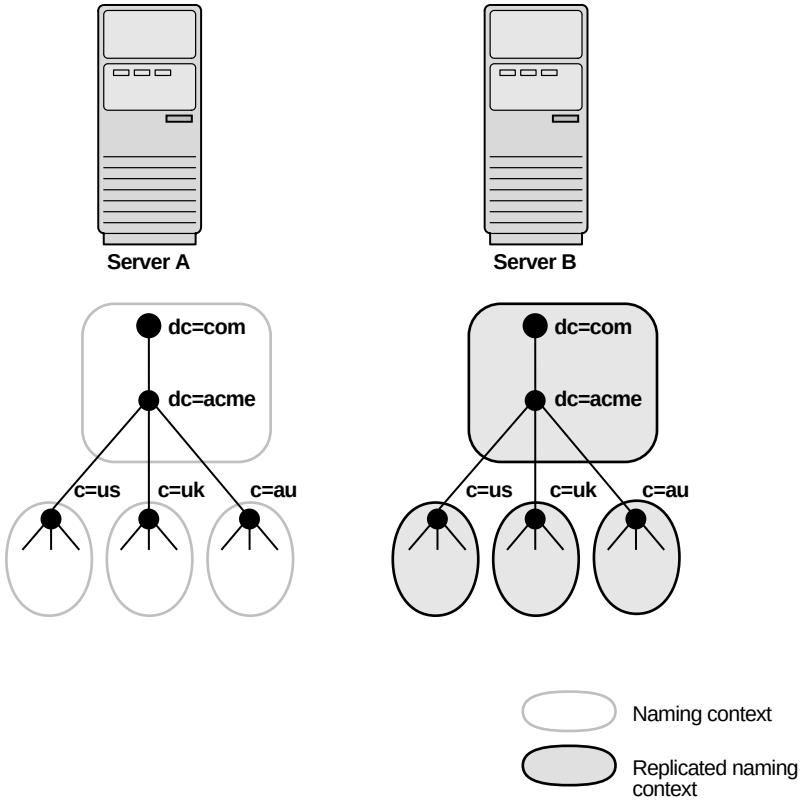
- Replication
- Partitioning

Replication

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

Figure 2–6 shows a replicated directory.

Figure 2–6 A Replicated Directory



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

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

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

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

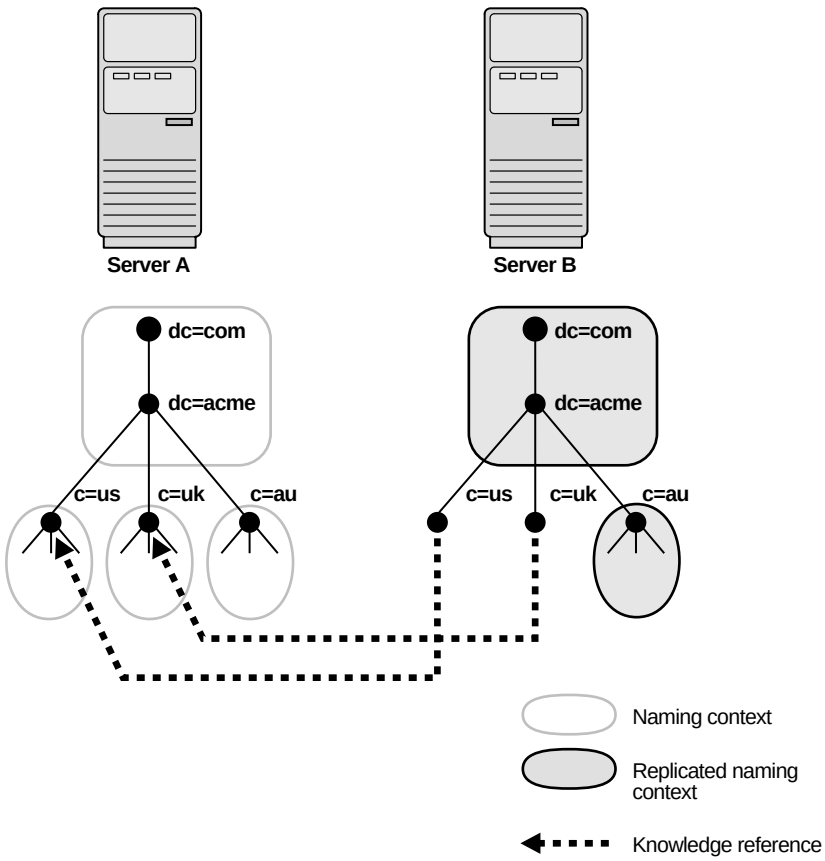
See Also: Chapter 22, "Directory Replication Concepts" for a more detailed discussion of replication, including: Oracle9*i* Replication architecture, change log purging, conflict resolution, and the replication process

Partitioning

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

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

Figure 2–7 A Partitioned Directory



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

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

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

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

The directory uses one or more **knowledge reference** to locate information that is requested of Server B, but that resides on Server A. It passes this information to a client in the form of a **referral.**

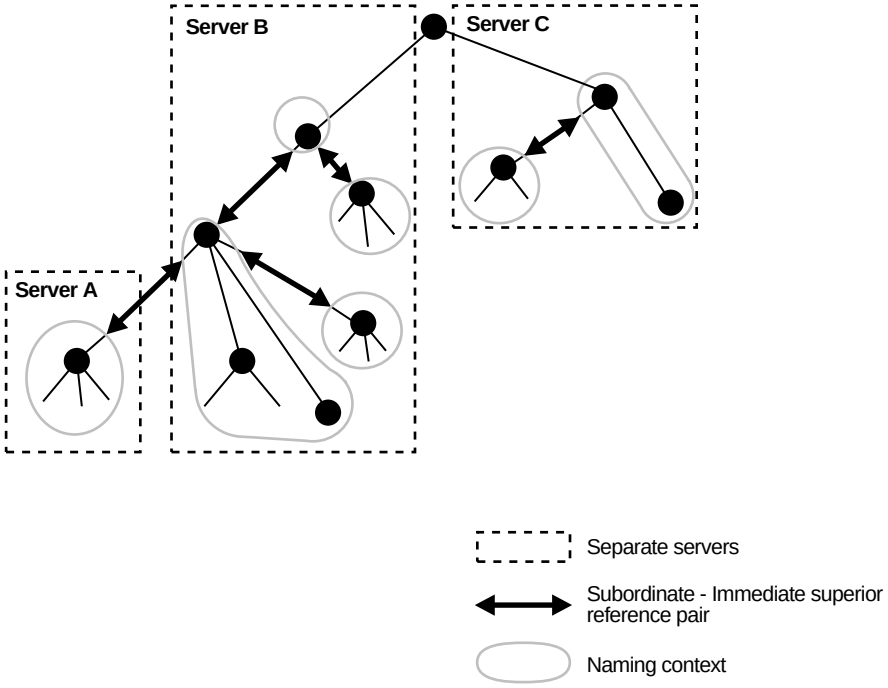
About Knowledge References and Referrals

A knowledge reference provides the names and addresses of the various naming contexts held in another partition. In Figure 2–7, Server B uses knowledge references to point to the `c=us` and `c=uk` naming contexts on Server A. When clients request from Server B the information residing on Server A, then Server B provides them with one or more referrals to Server A. Clients can then use these referrals to contact Server A.

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

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

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



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

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

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

Kinds of Referrals

There are two kinds of referrals:

- Smart referral

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

  For example, suppose that:

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

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

- Default referral

  Returned when the base object is not in the directory, and the operation is performed in a naming context on another server. A default referral typically sends the client to a server that has more knowledge about the directory partitioning arrangement.

  For example, suppose that Server A holds:

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

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

  See Also: "Managing Knowledge References and Referrals" on page 7-19

The Delegated Administration Service

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

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

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

The Oracle Directory Integration Platform

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

This section contains these topics:

- About Metadirectories
- About the Oracle Directory Integration Platform Environment

About Metadirectories

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

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

A metadirectory solves these problems by synchronizing information between all enterprise directories, forming one virtual directory. It centralizes administration,

thereby reducing administrative costs, and ensures that data is consistent and up-to-date across the enterprise.

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

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

About the Oracle Directory Integration Platform Environment

The Oracle Directory Integration Platform enables an enterprise to integrate its applications and other directories with Oracle Internet Directory. This platform provides all the interfaces and infrastructure necessary to keep the data in Oracle Internet Directory consistent with the data in enterprise applications and connected directories.

For example, an enterprise might want employee records in its HR database to be synchronized with Oracle Internet Directory. In addition, the enterprise may deploy certain LDAP-enabled applications (such as Oracle9*i*AS Portal) that need to be notified whenever changes are applied to Oracle Internet Directory. This service is called provisioning, and the Oracle Directory Integration Platform provides such applications with the necessary notifications.

Based on the nature of integration, the Oracle Directory Integration Platform provides two distinct services:

- The synchronization integration service, which keeps connected directories consistent with the central Oracle Internet Directory
- The provisioning integration service, which sends notifications to target applications periodically to reflect changes made to a user's status or information

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

3

Preliminary Tasks and Information

Before configuring and using Oracle Internet Directory, you must perform the tasks described in this chapter. This chapter also lists the locations of the log files of the various Oracle Internet Directory components.

This section contains these topics:

Task 1: Start the OID Monitor

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

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

This section contains these topics:

- Starting the OID Monitor
- Stopping the OID Monitor

Starting the OID Monitor

To start the OID Monitor:

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

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

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

For example:

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

Stopping the OID Monitor

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

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

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

For example:

```
oidmon connect=dbsl stop
```

Task 2: Start a Server Instance

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

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

This section contains these topics:

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

Starting an Oracle Directory Server Instance

The syntax for starting an Oracle directory server instance is:

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

Argument	Description
`connect=`*`net_service_name`*	If you already have a `tnsnames.ora` file configured, this is the net service name specified in that file, located in `ORACLE_HOME/network/admin`
`server=oidldapd`	Type of server to start (valid values are OIDLDAPD and OIDREPLD). This is not case-sensitive.
`instance=`*`server_instance_number`*	Instance number of the server to start. Should be a number between 1 and 1000.
`configset=`*`configset_number`*	Configset number used to start the server. This defaults to `configset0` if not set. This should be a number between 0 and 1000.
`-p` *`port_number`*	Specifies a port number during server instance startup. The default port number is 389.
`-work` *`maximum_number_of_worker_threads_per_server`*	Specifies the maximum number of worker threads for this server
`-debug` *`debug_level`*	Specifies a debug level during Oracle directory server instance startup
`-l` *`change_logging`*	Turns replication change logging on and off. To turn it off, enter `-l false`. To turn it on, do any one of the following: ■ omit the `-l` flag ■ enter simply `-l` ■ enter `-l true` Turning off change logging for a given node by specifying `-l false` has two drawbacks: it prevents replication of updates on that node to other nodes in the DRG, and it prevents application provisioning and synchronization of connected directories, because those two services require an active change log. The default, TRUE, permits replication, provisioning, and synchronization.

Argument	Description
`-server` *`number_of_server_processes`*	Specifies the number of server processes to start on this port
`start`	Starts the server specified in the `server` argument.

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

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

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

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

Single quotes are mandatory around the flags.

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

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

Stopping an Oracle Directory Server Instance

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

At the system prompt, type:

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

For example:

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

Starting an Oracle Directory Replication Server Instance

The syntax for starting the Oracle directory replication server is:

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

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

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

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

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

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

Single quotes are mandatory around the flags.

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

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

Stopping an Oracle Directory Replication Server Instance

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

At the system prompt, type:

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

For example:

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

Restarting Directory Server Instances

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

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

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

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

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

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

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

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

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

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

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

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

Troubleshooting Directory Server Instance Startup

If the directory server fails to start, you can override all user-specified configuration parameters to start the directory server and then return the configuration sets to a workable state by using the hard coded default parameters. Use this option only if the LDAP server fails to come up with default configset(`configset=0`).

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

```
oidctl connect=net_service_name server=oidldapd instance=1 flags='-p port_number
-f'
```

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

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

Task 3: Reset the Default Security Configuration

Oracle Internet Directory is installed with a default security configuration described later in this section. At the very beginning, you need to modify this default configuration to the needs of your environment, ensuring 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

Default Access Policies

When you first install Oracle Internet Directory, the default configuration allows the following policies at various points in the directory information tree.

Default Access Policy At the Root DSE

- Everyone has browse entry permissions
- The user security administration group and the user (self) have complete access to their own `userpkcs12`, `orcluserpkcs12hint`, `userpassword` `orclpassword`, and `orclpasswordverifier` attributes but not to those of others
- The user (self) has complete access to the user's own `orclpassword`, and `orclpasswordverifier` attribute but not to those of others
- Everyone has search, read, and compare access for all attributes except the `userpkcs12`, `orcluserpkcs12hint`, `userpassword`, `orclpassword`, and `orclpasswordverifier` attributes

Default Access Policy At the Users Container in the Default Subscriber Naming Context

The users container is `cn=users,o=oracle,dc=com`.

- The Subscriber DAS Create User Group (`cn=oracledascreateuser,cn=groups,cn=oraclecontext,`*`distinguished_name_of_subscriber`*) has permission to browse and add entries of the object class `orcluser`.
- The Subscriber DAS Delete User Group (`cn=oracledasdeleteuser,cn=groups,cn=oraclecontext,`*`distinguished_name_of_subscriber`*) has permission to browse and delete entries of the object class `orcluser`.
- The Subscriber DAS Edit User Group (`cn=oracledasedituser,cn=groups,cn=oraclecontext,`*`distinguished_name_of_subscriber`*) has permission to browse entries of object class `orcluser`.
- The Subscriber DAS Edit User Group has complete access to all attributes, including `userpassword`, in entries of the object class `orcluser`. The user (self) has complete access to the user's own attributes. Other users have only read permission on the attributes.
- The Authentication Services Group (`cn=authenticationServices,cn=groups,cn=oraclecontext,`*`distinguished_name_of_subscriber`*) has compare permission on `userpassword`, while other users have no permissions.
- The Verifier Services Group has read, search, and compare permission on `authpassword` and `orclpasswordverifier`. The user (self) has complete access to the user's own verifier attributes, while others have no access to them.

Default Access Policy At the Groups Container in the Default Subscriber Naming Context

The groups container is `cn=groups,`*`distinguished_name_of_subscriber`*`,cn=OracleContext`.

- The Subscriber DAS Create User Group has permission to browse and add entries of object class `orclgroup`.
- A hidden group entry of object class `orclgroup` can be added, deleted, or browsed only by the owner of that entry. Others have no permissions. Only the

owner has permissions to read, search, write, and compare attributes of such an entry.

- The owner of a public group entry of object class `orclgroup` can browse, add, and delete that entry. That entry can also be browsed by these groups:
  - DAS Create User Group
  - DAS Edit User Group
  - DAS Delete User Group

  Only the owner and the DAS Edit User Group have permission to read, search, write, and compare the attributes of such an entry.

Default Access Policy for the Oracle Context Administrators

The Oracle Context Administrators container is `cn=OracleContextAdmins,cn=groups,cn=OracleContext,` *`distinguished_name_of_subscriber`*. Members of the Oracle Context Administrators Group have complete administrative privileges over a specific Oracle Context. The have complete access to the Oracle Context in which the group exists.

Default Access Policy for Oracle9*i* Application Server Administrators

The Oracle9i Application Server Administrators container is `cn=IASAdmins,cn=groups,cn=OracleContext,`*`distinguished_name_of_subscriber`*. Members of the Oracle9*i* Application Server Administrators Group have complete administrative privileges over the Oracle9*i* Application Server product node in a given Oracle Context. In addition, they have permission to:

- Create application entity objects under individual products
- Proxy to these application entities

See Also:

- Chapter 2, "Concepts and Architecture" for an introduction to basic concepts, including security features of Oracle Internet Directory
- Chapter 4, "Directory Administration Tools" for information about the administration tools you use to configure security
- Chapter 13, "Directory Access Control" for a detailed explanation of access control options and instructions for setting up security
- Chapter 15, "Oracle Components and Oracle Internet Directory" for a detailed explanation of the Oracle Context schema
- Appendix C, "Schema Elements" for syntax and usage notes for the command-line tools

Task 4: Reset the Default Password for the Database

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

See Also: "OID Database Password Utility Syntax" on page A-49 for syntax and usage notes

Task 5: Run the OID Database Statistics Collection Tool

If you load data into the directory by any means other than the bulkload tool (bulkload.sh), then you must run the OID Database Statistics Collection tool after loading. Statistics collection is essential for the Oracle Optimizer to choose an optimal plan in executing the queries corresponding to the LDAP operations. You can run OID Database Statistics Collection tool at any time, without shutting down any of the OID daemons.

Note: To run this tool on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0.Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

See Also:

- "Using the OID Database Statistics Collection Tool" on page 4-15
- "OID Database Statistics Collection Tool Syntax" on page A-55

Log File Locations

The Oracle Internet Directory components output their log and trace information to log files in the *ORACLE_HOME* environment. Table 3–1 lists each component and the location of its corresponding log file.

Table 3–1

Component	Log File Name
Bulk Loader (bulkload.sh)	`$ORACLE_HOME/ldap/log/install.log`
Catalog Management Tool (catalog.sh)	`$ORACLE_HOME/ldap/log/catalog.log`
Directory integration agent	`$ORACLE_HOME/ldap/odi/log/`*`AgentName`*`.err` where *`AgentName`* is the name of the agent
Directory integration server (odisrv)	`$ORACLE_HOME/ldap/log/odisrv`*`XX`*`.log` where *`XX`* is Oracle directory integration server instance number
Directory replication server (oidrepld)	`$ORACLE_HOME/ldap/log/oidrepld00.log`
Directory server (oidldapd)	`$ORACLE_HOME/ldap/log/oidldapdXXs`*`pid`*`.log` where *`pid`* is the server process identifier
LDAP dispatcher (oidldapd)	`$ORACLE_HOME/ldap/log/oidldapd`*`XX`*`.log` where *`XX`* is the server instance number
OID Monitor (oidmon)	`$ORACLE_HOME/ldap/log/oidmon.log`

Table 3–1

Component	Log File Name
Replication setup (ldaprepl.sh)	`$ORACLE_HOME/ldap/admin/logs/ldaprepl.log`

4

Directory Administration Tools

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

This chapter contains these topics:

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

Directory administration is also aided by the Delegated Administration Service, which enables

- Delegated administrators, such as non-technical managers, to create and manage both users and groups
- End users to modify their own passwords without requiring the intervention of an administrator

See Also: Chapter 9, "The Delegated Administration Service"

Using Oracle Directory Manager

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

This section contains these topics:

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

---

Note: You cannot use Oracle Directory Manager to administer LDAP directories other than Oracle Internet Directory.

Starting Oracle Directory Manager

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

See Also:

- Chapter 3, "Preliminary Tasks and Information" for instructions on starting a server instance
- "Oracle Internet Directory Architecture" on page 2-15 for a conceptual explanation of directory server instances

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

Operating System	Instructions
Windows NT	From the Start menu, click Programs > *ORACLE_HOME* > Oracle Internet Directory > Oracle Directory Manager

Operating System	Instructions
Sun Solaris	If you have not set the path, then navigate to `ORACLE_HOME/bin`. Type at the system prompt: `oidadmin`

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

Connecting to a Directory Server

To connect to a directory server:

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

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

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

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

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

Field	Description
Password	If you are logging in as the super user and you specified a password for the super user during installation, in the Password box, type the password you specified. Otherwise, type the default password, namely, `welcome`. After you are logged into Oracle Directory Manager and have connected to a directory server, you should change this password to protect the directory. If you are logging in anonymously, leave the Password box empty. If you want to login as a specific directory user, enter the corresponding password. **See Also:** "Managing Super Users, Guest Users, and Proxy Users" on page 5-18 for instructions on how to change the password
Server	From the Server list, select the host containing the directory server to which you want to connect. If you are already connected to a directory server, and you want to connect to one on a different host: **1.** Click the button to the right of the Server field. The Select Directory Servers dialog box displays a list of available servers. **2.** Select a server. **3.** Click OK. To add a directory server to the list: **1.** In the Select Directory Servers dialog box, click Add. The Directory Server Connection dialog box appears. **2.** In the Server field, type the name of the directory server you want to add. **3.** In the Port field, type the port number for the server you want to add. **4.** Click OK. The added directory appears in the list in the Select Directory Server dialog box. To modify a directory server on the list: **1.** Select the directory server you want to modify. **2.** Click Edit. The Directory Server Connection dialog box appears. **3.** Modify the Server and Port fields, then click OK. The modifications for that server appear in the list in the Select Directory Server dialog box.

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

See Also:

- Chapter 12, "Secure Sockets Layer (SSL) and the Directory" for instructions on enabling SSL
- Appendix D, "Oracle Wallet Manager" for instructions on creating a wallet
- "Entries" on page 2-2 for instructions on formatting distinguished names
- "Configuring SSL Parameters" on page 12-3 for information about changing ports and their impact on security

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

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

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

5. Click Login. Oracle Directory Manager appears.

Navigating Oracle Directory Manager

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

Overview of Oracle Directory Manager

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

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

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

The Oracle Directory Manager Menu Bar

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

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

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

The Oracle Directory Manager Toolbar

Figure 4–1 and Table 4–1 together illustrate and describe the Oracle Internet Directory toolbar, starting at the left. Buttons become enabled or disabled depending on the pane or tab page you are displaying in Oracle Directory Manager.

Figure 4–1 Oracle Directory Manager Toolbar

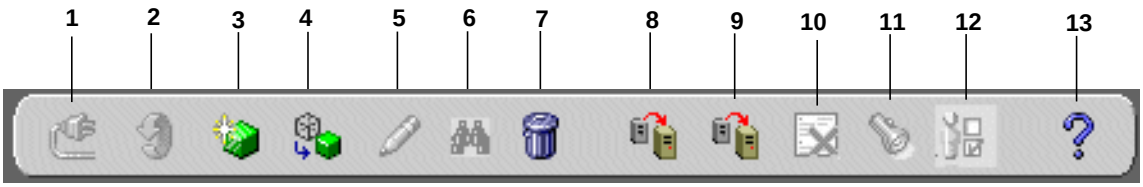


Table 4–1 Oracle Directory Manager Toolbar

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

Table 4–1 Oracle Directory Manager Toolbar

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

Connecting to Additional Directory Servers

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

To connect to an additional directory server:

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

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

Performing Administration Tasks by Using Oracle Directory Manager

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

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

Task Area	Instructions
Schema administration	"Managing Object Classes by Using Oracle Directory Manager" on page 6-6 "Managing Attributes by Using Oracle Directory Manager" on page 6-17
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 13-12 Managing Access Control by Using Command-Line Tools on page 13-42
Partitioning and replication	Chapter 23, "Oracle Directory Replication Server Administration"

Using Command-Line Tools

Oracle Internet Directory provides several types of command-line tools for manipulating directory entries and attributes:

- LDAP tools, for altering objects in text files written in the LDAP Data Interchange Format (LDIF)
- Bulk tools, for creating or managing large numbers of directory entries by using data from other applications
- A catalog management tool, for making existing attributes indexable

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

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

These three types of command-line tools are introduced in the subsections that follow, with references to detailed descriptions in an appendix.

Tools Affecting LDAP Entries Directly

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

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

See Also: "Using Globalization Support with Command-Line Tools" on page 8-5 for a discussion of command-line tools and Globalization Support

Using Bulk Tools

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

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

See Also:

- "Using the OID Database Password Utility" on page 4-14
- "OID Database Password Utility Syntax" on page A-49

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

Tool	Task(s)	Syntax and Usage Notes
bulkload	Load large number of entries to Oracle Internet Directory through LDIF files	"bulkload Syntax" on page A-35
ldifwrite	Copy data from the directory information base into an LDIF file that can be read by any LDAP compliant directory server. You can use ldifwrite in conjunction with bulkload. You can also use ldifwrite to back up information from all or part of a directory.	"ldifwrite Syntax" on page A-39
bulkmodify	Modify a large number of existing entries efficiently	"ldapmodify Syntax" on page A-15
bulkdelete	Delete a subtree efficiently	"bulkdelete Syntax" on page A-34

Using the Catalog Management Tool

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

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

See Also:

- "Catalog Management Tool Syntax" on page A-40 for syntax and usage notes
- "Indexing an Attribute by Using Command-Line Tools" on page 6-31
- "Indexing an Attribute by Using Oracle Directory Manager" on page 6-28

Using OID Control Utility

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

See Also:

- "OID Control Utility Syntax" on page A-43
- "Oracle Internet Directory Architecture" on page 2-15 for a conceptual description

Using the OID Database Password Utility

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

See Also: "OID Database Password Utility Syntax" on page A-49 for syntax and usage notes

Using the Replication Tools

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

At this point, you need to:

1. Examine the change in the human intervention queues
2. Reconcile the conflicting changes

3. Place the change either back into the retry queue or into the purge queue.

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

See Also:

- "Using the OID Reconciliation Tool" on page 23-31
- "OID Reconciliation Tool Syntax" on page A-52 for syntax and an explanation of how OID Reconciliation Tool works
- "Using the Human Intervention Queue Manipulation Tool" on page 23-31
- "Human Intervention Queue Manipulation Tool Syntax" on page A-49

Using the OID Database Statistics Collection Tool

The OID Database Statistics Collection tool (oidstats.sh) is located in `$ORACLE_HOME/ldap/admin`. You must run this utility whenever there are significant changes in directory data—including the initial load of data into the directory.

If you load data into the directory by any means other than the bulkload tool (bulkload.sh), then you must run the OID Database Statistics Collection tool after loading. Statistics collection is essential for the Oracle Optimizer to choose an optimal plan in executing the queries corresponding to the LDAP operations. You can run OID Database Statistics Collection tool at any time, without shutting down any of the OID daemons.

Note: To run this tool on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

See Also: "OID Database Statistics Collection Tool Syntax" on page A-55

Administration Tasks at a Glance

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

Task	Information
Managing Attributes	
Add, modify, or delete an attribute by using command-line tools	"Managing Attributes by Using Command-Line Tools" on page 6-29
Add, modify, or delete an attribute by using the Oracle Directory Manager	"Managing Attributes by Using Oracle Directory Manager" on page 6-17
Managing Entries	
Add, modify, or delete a directory entry by using command-line tools	"Managing Entries by Using Command-Line Tools" on page 7-13
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 Syntax" on page A-35 "LDAP Data Interchange Format (LDIF) Syntax" on page A-2
View Directory Information Tree (DIT) hierarchy of entries	"Managing Entries by Using Oracle Directory Manager" on page 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 6-14
Add, modify, or delete object classes by using Oracle Directory Manager	"Managing Object Classes by Using Oracle Directory Manager" on page 6-6
Managing Replication	
Set up replication	Chapter 23, "Oracle Directory Replication Server Administration"
Resolve replication change conflicts	"Resolving Conflicts Manually" on page 23-29
Move replication changes from human intervention queue to either the retry queue or the purge queue	"Using the Human Intervention Queue Manipulation Tool" on page 23-31
Managing Security	
Set up an Access Control Policy Point (ACP)	Chapter 13, "Directory Access Control"

Task	Information
Set up SSL	Chapter 12, "Secure Sockets Layer (SSL) and the Directory"
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 4-10
Start the directory server processes	Chapter 3, "Preliminary Tasks and Information"
Stop the directory server processes	Chapter 3, "Preliminary Tasks and Information"
View system operational attributes	"Setting System Operational Attributes by Using Oracle Directory Manager" on page 5-13 "Setting System Operational Attributes by Using ldapmodify" on page 5-16

Part II

Basic Directory Administration

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

5

Oracle Directory Server Administration

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

This chapter contains these topics:

- Managing Server Configuration Set Entries
- Setting System Operational Attributes
- Managing Naming Contexts
- Managing Super Users, Guest Users, and Proxy Users
- Configuring Searches
- Monitoring, Debugging, and Auditing the Directory Server
- Viewing Active Server Instance Information
- Changing the Password to an Oracle Database Server
- Dereferencing Alias Entries

See Also: Chapter 3, "Preliminary Tasks and Information" for instructions on starting and stopping directory server instances

Managing Server Configuration Set Entries

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

See Also:

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

This section contains these topics:

Preliminary Considerations for Managing Configuration Set Entries

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 this applies to the Oracle directory server instances only. The Oracle replication directory server supports only one configuration set.

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

Figure 5–1 shows three separate directory server instances, each with a different value.

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

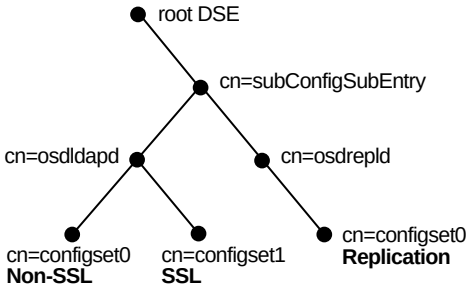


Figure 5–1 shows:

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

See Also:

- Chapter 12, "Secure Sockets Layer (SSL) and the Directory" for information about configuration parameters for SSL
- Chapter 23, "Oracle Directory Replication Server Administration" for information about configuration parameters for replication
- "Configuration Set Entry Attributes" on page C-5 for a list and descriptions of the entire set of attributes that are used to configure an instance of a directory server

Managing Server Configuration Set Entries by Using Oracle Directory Manager

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

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

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

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

Viewing Configuration Set Entries by Using Oracle Directory Manager

To view configuration set entries:

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

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

   See Also: "Modifying Configuration Set Entries by Using Oracle Directory Manager" on page 5-8

Adding Configuration Set Entries by Using Oracle Directory Manager

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

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

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

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

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

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

Field	Description
SSL Enable	Set 0 for only non-secure operation; default port is 839, changeable below. Set 1 for only SSL authentication; default port is 636, changeable below. Set 2 for both non-secure operation and SSL authentication.

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

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

SSL Wallet Password	Type the password for the server-side wallet. This password was set during creation of the wallet. If you change the password, you must change this parameter.
SSL Wallet Confirm Password	Retype the new password in this field when you change the password.
SSL Port	The default SSL port is 636. You can change the SSL port.
Non-SSL Port	The default non-SSL port is 839. You can change the non-SSL port.

5. Click Apply.

Note: Remember: The changes will not affect the active directory server instance until you restart it. See "Restarting Directory Server Instances" on page 3-7.

See Also:

- Appendix D, "Oracle Wallet Manager" for information about setting the location of the Oracle Wallet and the Oracle Wallet password
- "Setting Debug Logging Levels by Using the OID Control Utility" on page 5-27

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

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

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

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

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

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

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

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

4. Click Ok.

Modifying Configuration Set Entries by Using Oracle Directory Manager

To modify configuration set entries:

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

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

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

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

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

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

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

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

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

Note: Remember: The changes will not affect the active directory server instance until you restart it. See "Restarting Directory Server Instances" on page 3-7.

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

Deleting Configuration Set Entries by Using Oracle Directory Manager

To delete configuration set entries:

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

Note: Remember: The changes will not affect the active directory server instance until you restart it. See "Restarting Directory Server Instances" on page 3-7.

Managing Server Configuration Set Entries by Using Command-Line Tools

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

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

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

Adding Configuration Set Entries by Using ldapadd

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

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

1. Create the input file in a text editor.

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

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

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

2. Run ldapadd with an input file.

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

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

   See Also:

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

Modifying and Deleting Configuration Set Entries by Using ldapmodify

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

1. Create the input file.

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

   Input files must have LDIF format.

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

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

2. Run ldapmodify referencing the input file.

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

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

See Also:

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

Setting System Operational Attributes

An operational **attribute**—as opposed to an application attribute—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. You must have superuser privileges to set system operational attributes.

This section contains these topics:

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

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

Setting System Operational Attributes by Using Oracle Directory Manager

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

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

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

Field	Description	Default Value	Modifiable?
Server Mode	Determines whether data can be written to the server. You can change this value to either Read / Write or Read Only. Change the default to Read Only during replication process.	Read / Write	Choices are Read / Write and Read-Only
Server Operation Time Limit	Maximum amount of time, in seconds, allowed for a search to be completed	3600	Yes
Supported Control	Extension information for any LDAP operation. The control types supported by Oracle Internet Directory are listed as values of the `supportedcontrol` attribute in the root DSE. Each control type has an associated object identifier defined by the LDAP standard.The values of the supportedcontrol attribute are standard object identifiers assigned to control types.	`manageDSACtrl`	No
orclecacheenabled	Specify whether entry caching is enabled. The value for enabled is 1; the value for disabled is `0`.	1	Yes
orclecachemaxsize	Specify the maximum number of bytes of RAM that the entry cache can use.	100M	Yes
orclecachemaxentries	Specify the maximum number of entries that can be present in the entry cache.	25,000	Yes

Setting System Operational Attributes by Using ldapmodify

The modifiable system operational attributes are:

Attribute	Description	Default
`namingContexts`	Topmost DNs for the naming contexts contained in this server. You must have super user privileges to publish a DN as a naming context.	none
`orclCryptoScheme`	Hash algorithm for encrypting the password. Options are: ■ MD4 ■ MD5 ■ No encryption ■ SHA ■ UNIX Crypt	MD4
`orclSizeLimit`	Maximum number of entries to be returned by a search	1000
`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
orclecacheenabled	Specification as to whether entry caching is enabled. The value for enabled is 1; the value for disabled is `0`.	1
orclecachemaxsize	Maximum number of bytes of RAM that the entry cache can use.	100M
orclecachemaxentries	Maximum number of entries that can be present in the entry cache.	25,000

Note: Entry caching is automatically disabled in multiserver OID instances, irrespective of the value of orclecacheenabled.

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

Managing Naming Contexts

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

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

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

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

Before you publish a naming context, be sure that:

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

This section contains these topics:

Publishing Naming Contexts by Using Oracle Directory Manager

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

Publishing Naming Contexts by Using ldapmodify

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

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

Managing Super Users, Guest Users, and Proxy Users

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

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

A **proxy user**, as described in "Indirect Authentication" on page 11-5, is typically used in an environment with a middle tier such as a firewall or a RADIUS server. The default user name for a proxy user is `proxy`; the default password is `proxy`.

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

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

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

This section contains these topics:

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

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

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

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

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

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

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

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

Managing Super, Guest, and Proxy Users by Using ldapmodify

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

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

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

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

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

Configuring Searches

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

This section contains these topics:

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

Configuring Searches by Using Oracle Directory Manager

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

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

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

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

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

Configuring Searches by Using ldapmodify

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

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

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

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

Setting the Maximum Amount of Time For Searches by Using ldapmodify

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

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

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

Monitoring, Debugging, and Auditing the Directory Server

This section contains these topics:

- Monitoring Oracle Internet Directory Servers by Using Oracle Internet Directory Server Manageability Framework
- Setting Debug Logging Levels
- Using the Audit Log

Monitoring Oracle Internet Directory Servers by Using Oracle Internet Directory Server Manageability Framework

The Oracle Internet Directory Server Manageability framework enables you to monitor the following directory server statistics:

- Server health statistics about LDAP request queues, memory, LDAP sessions, and database sessions
- General statistics about server operations and queues
- Critical events related to security
- Critical events related to system resources
- Status information of the replication server
- Status information of Oracle directory integration server and the integration profiles.

See Also: Chapter 28, "Oracle Directory Integration Platform Concepts and Components"

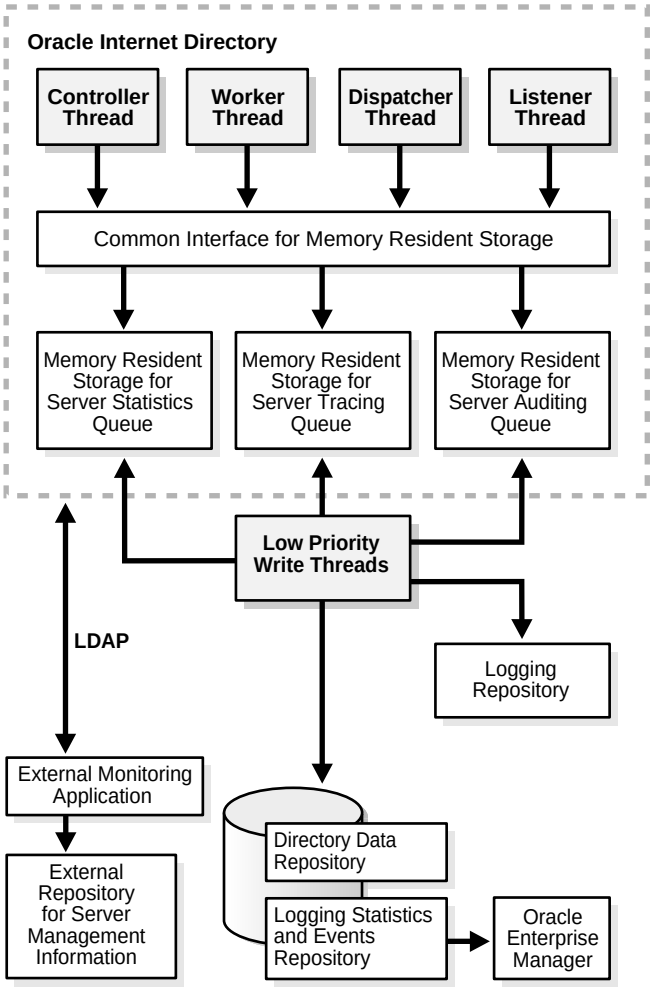
Please note that no LDAP query interface is provided for these statistics and events.

Some monitored information is made available by means of a web-based Oracle Enterprise Manager GUI tool. Please refer to the online Help for the Oracle Enterprise Manager web-based GUI tool for OID Server manageability.

Oracle Internet Directory Server Manageability Architecture and Components

Figure 5–2 and the accompanying text explain the relationship between the various components of directory server manageability.

Figure 5–2 Architecture of Oracle Internet Directory Server Manageability



Directory Server A directory server responds to directory requests from clients. It has four kinds of functional threads: dispatcher, listener, controller, and worker. It accepts LDAP requests from clients, processes them, and sends the LDAP response back to the clients.

When you use the Oracle Internet Directory Server Manageability framework to set runtime monitoring, the four functional threads of the server record the specified information and store it in local memory.

> **See Also:** "An Oracle Directory Server Instance" on page 2-20 for a description of the directory server

Memory Resident Storage This is a local process memory. The Oracle Internet Directory Servers Manageability framework assigns one each for statistics, tracing, and auditing. Each has its own separate data structure maintained in the local memory storage.

Low-Priority Write Threads These dedicated write threads differ from server functional threads in that they write server statistics, audit logging, and tracing information to the repository. To maintain reduced system overhead, their priorities are kept low.

External Monitoring Application This module, which is proprietary and external to the server manageability framework, collects the gathered statistics through a standard LDAP interface with the directory server and stores it in its own repository.

External Repository for Server Management Information This is the repository that the monitoring agent uses to store the gathered directory server statistics. The monitoring agent determines how this repository is implemented.

Oracle Enterprise Manager (OEM) Oracle Enterprise Manager extracts monitored data from the statistics and events repository, presenting it in a Web-based graphical user interface. Users can view the data in a normal browser. A repository can store the collected data for generic and custom queries.

Logging Repository (File System) This repository uses a file system to store information traced across various modules of the directory server. By using a file system for this purpose, the Oracle Internet Directory Server Manageability framework uses the features and security of the operating system.

Directory Data Repository This repository contains all user-entered data—for example, user and group entries.

Statistics and Events Repository This repository is like the tracing repository except that it stores the information in the same database as the directory data repository rather than in a file system. In this way, the Oracle Internet Directory Server Manageability framework uses:

- Normal LDAP operations to store and retrieve the information
- Existing access control policies to manage the security of the gathered information

The directory manageability framework isolates the gathered information from the directory data by storing the two separately.

Location of Configuration Information for Oracle Internet Directory Server Manageability

The Oracle Internet Directory Server Manageability framework stores configuration parameters in the DSE root of the directory for all three modules—namely, server statistics, server tracing, and server auditing. To specify periodicity, amount, and level of information to be gathered, you must set appropriate values for these parameters.

Configuring Server Manageability

To configure the Oracle Internet Directory Server Manageability framework, use ldapmodify to set positive integer values for these attributes in the DSE root entry:

Attribute	Description
orclStatsFlag	Indicate whether you want to enable or disable the Oracle Internet Directory Server Manageability framework. To enable, set this to `1`. To disable, set it to `0`.
orclStatsPeriodicity	Specify how often you want to gather sample statistics—that is, the number of minutes in the interval. Set this to 1 or more minutes.

See Also: Oracle Enterprise Manager online help for more information about monitoring and managing Oracle Internet Directory servers by using server manageability

Setting Debug Logging Levels

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

This section contains these topics:

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

Setting Debug Logging Levels by Using Oracle Directory Manager

To set the debug logging level:

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

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

Setting Debug Logging Levels by Using the OID Control Utility

To set debug logging levels by using the OID Control Utility, restart the Oracle directory server using the `-debug` flag for an LDAP server, and the `-d` flag for the replication server. Use the debug level number based on Table 5–1.

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

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

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

Table 5–1 provides the complete list of debug logging levels.

Table 5–1 Debug Logging Levels

Logging Level Value	Provides Information Regarding
1	Trace function calls
2	Debug packet handling
4	Heavy trace debugging (more information than level 1)
8	Connection management, related to network activities
16	Packets sent and received between server and client
32	Search filter processing
64	Configuration file processing
128	Access control list processing
256	Log of operations and results for each connection
512	Log of entries sent
1024	Log of communication with the back-end, i.e., with the database
2048	Entry parsing
4096	Schema-related operations
32768	Replication-specific operations
65535	All possible debugging operations / data

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

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

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

Using the Audit Log

The audit log records critical events on the Oracle directory server that are important from both a security and an operational point of view. Because the log generation depends on events on the directory server, you cannot create audit log entries. Only the directory server itself can create them.

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

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

See Also:

- "Auditable Events" on page 5-31 for a listing of audit levels
- "Setting the Audit Level" on page 5-32 for instructions on specifying the audit level
- "Searching for Audit Log Entries by Using Oracle Directory Manager" on page 5-33
- "Searching for Audit Log Entries by Using ldapsearch" on page 5-35
- "ldapdelete Syntax" on page A-11

Structure of Audit Log Entries

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

Attribute	Description
`orclsequence`	Used to create the name of the entry. The name is generated using a database sequence.
`orcleventtype`	Specifies the type of event that occurred. This is a cataloged attribute.
`orcleventtime`	Specifies the time at which the event occurred. This is formatted in **UTC (Coordinated Universal Time)**. UTC is indicated by a z at the end of the value. For example, `orcleventtime: 199811281010z`

Attribute	Description
orcluserdn	Specifies the identity of the user who logged into the Oracle directory server to perform the operation. This attribute is cataloged.
orclopresult	Specifies the outcome of the operation. It states either SUCCESS if the operation succeeds, or the reason why the operation failed.
orclauditmessage	Specifies the textual message. This attribute is not cataloged.
objectclass	Contains the preset values top and orclauditoc.

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

Note: By default, the attributes orcleventtype and orcluserdn are indexed at installation of Oracle Internet Directory. If you drop the indexes from these attributes, you cannot search for them. To re-create the index for these attributes, use the Catalog Management tool. See "Indexing an Attribute by Using Command-Line Tools" on page 6-31.

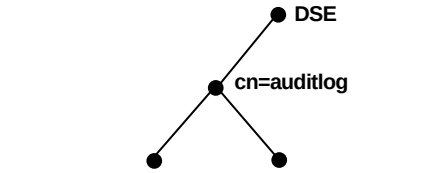
See Also:

- "Catalog Management Tool Syntax" on page A-40 for information about cataloged attributes
- "Object Class Types" on page 2-10 for a description of top

Position of Audit Log Entries in the DIT

The audit log container is part of the DSE. It holds its entries as children, organized according to the orclsequence attribute. See Figure 5–3.

Figure 5–3 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 5–2 shows the auditable events and their audit levels. The third column, Audit Levels, contains hexidecimal values. You can audit more than one event by adding their corresponding values found in this column.

Table 5–2 Auditable Events

Event	Description	Audit Levels
Superuser login	Super user bind to the server (successes or failures)	0x0001
Schema element add / replace	Addition of a new schema element (successes or failures)	0x0002
Schema element delete	Deletion of a schema (successes or failures)	0x0004
Bind	Unsuccessful bind cases	0x0008
Access violation	Access denied by **access control policy point**	0x0010
directory-specific entry (DSE) modification	Changes to a DSE (successes or failures)	0x0020
Replication login	Replication server authentication (successes or failures)	0x0040
ACL modification	Changes to an **access control list (ACL)**	0x0080

Table 5–2 Auditable Events

Event	Description	Audit Levels
User password modification	Modification of user password attribute	0x0100
Add	ldapadd operation (successes or failures)	0x0200
Delete	ldapdelete operation (successes or failures)	0x0400
Modify	ldapmodify operation (successes or failures)	0x0800
ModifyDN	ldapModifyDN operation (successes or failures)	0x1000

Setting the Audit Level

The setting for the DSE attribute `orclauditlevel` indicates the current audit level. You can enable or disable the events described in the previous section. A value of `0` for this attribute, which is the default, disables auditing.

You can set the audit level by using either Oracle Directory Manager or ldapmodify. This section describes both methods.

Setting the Audit Level by Using Oracle Directory Manager To set the audit level by using Oracle Directory Manager:

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

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

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

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

> **See Also:** "Restarting Directory Server Instances" on page 3-7 for instructions on how to restart the directory server

> **See Also:** "Auditable Events" on page 5-31 for a description of each audit level

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

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

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

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

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

> **See Also:** "Restarting Directory Server Instances" on page 3-7 for instructions on how to restart the directory server

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

To use Oracle Directory Manager to view audit log entries:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*, and select Audit Log Management. The corresponding right pane appears.

2. In the Max Results (entries) field, type the maximum number of entries you want your search to retrieve. The default is 200. The directory server retrieves the number you specify, up to 1000.
3. In the Max Search Time (seconds) box, type the maximum number of seconds for the duration of your search. The value you enter here must be at least that of the default, namely, 25. The directory server searches for the amount of time you specify, up to one hour.
4. In the Search Criteria box, use the lists and text fields on the search criteria bar to focus your search.
   a. From the list at the left end of the search criteria bar, select an attribute of the entry for which you want to search. Because not all attributes are used in every entry, be sure that the attribute you specify actually corresponds to one in the entry for which you are searching. Otherwise, the search fails.
   b. From the list in the middle of the search criteria bar, select a filter. Options are:

Filter	Description
Begins With	Searches by using only the first few characters of the attribute's value.
Ends With	Searches for an entry by using only the last few characters of the specified attribute's value.
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter.
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter.
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship.

   c. In the text box at the right end of the search criteria bar, type the value for the attribute you just selected. For example, if the attribute you selected was cn, you could type the particular common name you want to find.

5. To further refine your search, use the buttons in the Search Criteria box to enhance the search criteria bar.

Button	Description
New	Creates a new search criteria bar in the Search Criteria field. This button is enabled only when the Search Criteria field is empty.
And	Creates another search criteria bar in the Search Criteria field. Matches all entries with one specified attribute with those that also have another specified attribute. For example, `cn=Baldwins And title=Laborer` retrieves all Baldwins who are also laborers.
Or	Creates another search criteria bar in the Search Criteria field. Matches all entries with either one specified attribute or another. For example, `title=Laborer Or title=Foreman` retrieves all employees who are either laborers or foremen.
Not	Negates the criterion in the selected search criteria bar and retrieves all entries that do not have the specified criterion. For example, `cn=Frank And Not title=Laborer` retrieves all persons named Frank who are not laborers.
Delete	Deletes a selected search criteria bar

6. Click Search. The results of your search appear in the Distinguished Name box.
7. To view the properties of a particular audit log entry, select it in the Distinguished Name box, then click View Properties. The Audit Log Entry dialog box displays the properties for the audit log entry you selected.

See Also: "Configuring Searches" on page 5-20 for instructions on setting the number of entries to display in searches, and to set the time limit for searches

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

See: "ldapsearch Syntax" on page A-22

Purging the Audit Log

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

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

Viewing Active Server Instance Information

To view information about any active directory server instance—including type, instance number, debug level, host name, and configuration parameters—use **Oracle Directory Manager**. To do this:

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

> **See Also:** "Managing Server Configuration Set Entries by Using Oracle Directory Manager" on page 5-4 for instructions on changing configuration set entries

Changing the Password to an Oracle Database Server

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

> **See Also:** "OID Database Password Utility Syntax" on page A-49

Dereferencing Alias Entries

This section explains concepts about dereferencing alias entries, the usage model, and includes a list of messages.

This section contains these topics:

Concepts for Dereferencing Alias Entries

Alias entries in the LDAP directory enable one entry to point to another entry, so you can devise structures that are not strictly hierarchical. Alias entries perform a function like symbolic links in the UNIX file system or shortcuts in the Windows 95/NT file system.

In Figure 5–4, the `ou=uk sales,ou=global sales,o=oracle,c=us` entry is an alias entry pointing to the `ou=sales,o=oracle,c=uk` entry. The pointer (like all information) is held as an attribute, the aliased object name attribute of the alias entry. Alias entries have special object class alias to distinguish them from object entries in a directory.

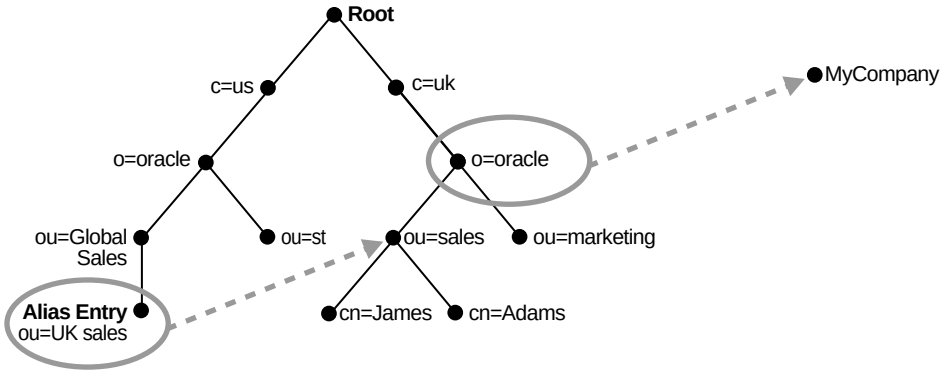
Alias Objectclass Definition

```
(2.5.6.1 NAME 'alias' SUP top STRUCTURAL MUST aliasedObjectName)
```

Aliased Objectname Definition

```
(2.4.5.1 NAME 'aliasedObjectName" EQUALITY distinguishedNmameMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE)
```

Figure 5–4 Alias Entries Example



Anyone referencing `ou=uk sales,ou=global sales,o=oracle,c=us` is automatically rerouted to the real entry `ou=sales,o=oracle,c=uk` by the LDAP server. This process is called alias dereferencing.

Using Alias Entry Dereferencing

This section contains these topics:

- Adding an Alias Entry
- Searching the Base
- Searching One-Level
- Searching a Subtree
- Modifying Alias Entries

Adding an Alias Entry

Use the following LDIF to create a normal entry and an alias entry pointing to the real entry. When you add the information in the steps, the tree in Figure 5–5 is the result.

1. Create a sample.ldif file with the following entries:

```
dn: c=us
c: us
objectclass: country
```

```
dn: o=oracle, c=us
o: oracle
objectclass:organization

dn: ou=Area1, c=us
objectclass: alias
aliasedObjectName: o=oracle, c=us

dn: cn=John Doe, o=oracle, c=us
cn: John Doe
objectclass: person

dn: cn=President, o=oracle, c=us
objectclass: alias
aliasObjectName:  cn=John Doe, o=oracle, c=us
```

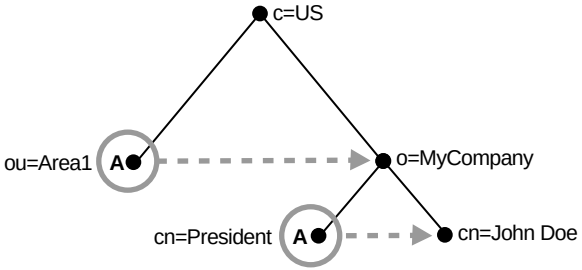
2. Add entries to the directory using the following command:

```
ldapadd -p <port> -h <host> -f  sample.ldif
```

Note: When you add an alias entry whose parent is an alias entry, the LDAP server will return an error.

See Also: Entry Alias Dereferencing Messages on page 5-43 for error messages

Figure 5–5 Resulting Tree when Creating the sample.ldif File



In Figure 5–5, the letter A represents an alias entry, where:

- `ou=Area1` is an alias pointing to `o=oracle`
- `cn=President` is an alias pointing to `cn=John Doe`

Searching the Base

A base search finds the top-most level of the alias entry you specify.

For example, perform a base search of `"ou=Area1,c=us"` with a filter of `"objectclass=*"` with the `-deref` option `LDAP_DEREF_FINDING` as follows:

```
ldapsearch -p <port> -h <host> -b "ou=Area1,c=us"  -a find -s base
“objectclass=*”
```

The directory server, during the base search, looks up the base specified in the search request and returns it to the user if located. If the base is an alias entry and, as in the example, `-a find` is specified in the search request,then the LDAP server automatically dereferences the alias entry and returns the dereferenced entry. Therefore, the search dereferences `ou=Area1,c=us` (which is an alias entry) and `o=oracle,c=us` is returned.

Searching One-Level

A one-level search finds only the child to the base level you specify.

In each search you specify, there are flags you can set. The search is performed based on the flag you specify.

The flags are as follows:

Flag	Content
`LDAP_DEREF_NEVER`	`-a never`
`LDAP_DEREF_FINDING`	`-a find`

By default, the dereference flag in `ldapsearch` is `LDAP_DEREF_NEVER` (that is, `-a never`) and thus the LDAP server does not perform any dereferencing for alias entries.

For example, perform a one-level search of "ou=Area1,c=us" with a filter of "objectclass=*" with the -deref option set to LDAP_DEREF_FINDING (-a find) as follows:

```
ldapsearch -p <port> -h <host> -b "ou=Area1,c=us"  -a find -s one
"objectclass=*"
```

The search operation is performed by the LDAP server in two steps.

1. The LDAP server searches for the base that is specified in the search request.
2. When the LDAP server locates the base, then it looks up all one-level entries under this base and returns entries that match the filter criteria.

In the example, -a find is specified in the search request, thus the LDAP server automatically dereferences while looking up the base (the first step), but does not dereference alias entries that are one level under the base. Therefore, the search dereferences ou=Area1,c=us (which is an alias entry) and then looks up one-level entries under o=oracle,c=us. One of the one-level entries is cn=President,o=oracle,c=us that is not dereferenced and is returned as is.

Thus, the search returns cn=President,o=oracle,c=us and cn=John Doe,o=oracle,c=us.

Searching a Subtree

A subtree search finds the base, children, grand children, (the family tree).

In each search you specify, there are flags you can set. The search is performed based on the flag you specify.

The flags are as follows:

Flag	Content
LDAP_DEREF_NEVER	-a never
LDAP_DEREF_FINDING	-a find

By default, the dereference flag in ldapsearch is LDAP_DEREF_NEVER (that is, -a never) and thus the LDAP server does not perform any dereferencing for alias entries.

For example, perform a subtree search of "ou=Area1,c=us" with a filter of "objectclass=*" with the -deref option LDAP_DEREF_FINDING as follows:

```
ldapsearch -p <port> -h <host> -b "ou=Area1,c=us"  -a find -s one
"objectclass=*"
```

The search operation is performed by the LDAP server in two steps.

1. The LDAP server searches for the base that is specified in the search request.
2. When the LDAP server locates the base, then it looks up all entries under this base and returns entries that match the filter criteria.

In the example, -a find is specified in the search request, thus the LDAP server automatically dereferences while looking up the base (the first step), but does not dereference alias entries that are under the base. Therefore, the search dereferences ou=Area1,c=us (which is an alias entry) and then looks up entries under o=oracle,c=us. One of the entries is cn=President,o=oracle,c=us that is not dereferenced and is returned as is.

Thus, the search returns the following:

- o=oracle,c=us
- cn=john doe,o=oracle,c=us
- cn=President,o=oracle,c=us

Modifying Alias Entries

You can modify alias entries.

For example, create a sample.ldif file with following entries:

```
dn: cn=President, o=oracle, c=us
changetype : modify
replace: aliasObjectName
aliasObjectName:  cn=XYZ, o=oracle, c=us
```

Modify the alias entry using the following command:

```
ldapmodify -p <port> -h <host> -f sample.ldif
```

Success and Error Messages

The following messages are returned when encountering the alias issue in the description column.

Table 5–3 Entry Alias Dereferencing Messages

Message	Description
Alias Problem	The error message returns to the client when either of the following occur: An alias was dereferenced, but it did not point to an entry in the DIT. The user tries to add an alias entry whose parent is an alias.
Alias Dereferencing Problem	The error message returns to the client when the user is not allowed to dereference an alias because of access control issues.
No Such Object	The error message returns to the client when the server is not able to find the base DN specified in the search request.
Invalid DN Syntax	When adding or modifying an alias entry, if the value specified for `aliasedObjectName` has invalid DN syntax, then the LDAP server returns an `invalidDNSyntax` error message to the client.
Success	The LDAP server returns a success message when the client operation successfully completes. When the dereferenced target does exist but does not match the filter specified in the search request, the server returns a success message with no matched entry.
Insufficient Access Rights	The error message returns if the user does not have access to the dereferenced entry.

6

Directory Schema Administration

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

This chapter contains these topics:

About the Directory Schema

A directory schema does the following:

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

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

This chapter discusses each of these elements.

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

About Object Class Management

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

See Also:

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

This section contains these topics:

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

Guidelines for Adding Object Classes

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

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

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

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

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

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

See Also: "About Attribute Management" on page 6-16

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

When you add or perform an operation on an entry, you do not need to specify the entire hierarchy of superclasses associated with that entry. This feature, called object

class explosion, enables you to specify only the leaf object classes. Oracle Internet Directory resolves the hierarchy for the leaf object classes and enforces the information model constraints. For example, the `inetOrgPerson` object class has `top`, `person` and `organizationalPerson` as its superclasses. When you create an entry for a person entry, you need to specify only `inetOrgPerson` as the object class. Oracle Internet Directory then enforces the schema constraints defined by the respective superclasses, namely, `top`, `person`, and `organizationalPerson`.

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

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

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

Guidelines for Modifying Object Classes

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

You can make these changes to an object class:

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

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

- You cannot modify an object class that is part of the standard LDAP schema. You can, however, modify user-defined object classes. Also, if existing object

classes do not have the attributes you need, you can create an auxiliary object class and associate the needed attributes with it.

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

See Also:

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

Guidelines for Deleting Object Classes

There are also some limitations on deleting object classes:

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

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

Managing Object Classes by Using Oracle Directory Manager

This section contains these topics:

Searching for Object Classes by Using Oracle Directory Manager

You can specify your search for an object class by:

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

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

To search for an object class:

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

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

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

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

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

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

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

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

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

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

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

Viewing Properties of Object Classes by Using Oracle Directory Manager

To view all object classes in the schema:

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

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

3. In the Object Class dialog box:
   - Object classes from which attributes may be inherited are listed in the Super Class box
   - Mandatory attributes are listed in the Mandatory Attributes box
   - Optional attributes are listed in the Optional Attributes box

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

Adding Object Classes by Using Oracle Directory Manager

To add object classes by using Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server*, then select Schema Management.
2. Choose one of the following methods:
   - In the right pane, select the Object Classes tab and click the Create button in the toolbar.
   - Click the Create button at the bottom of the right pane.
   - From Operations menu, select Create Object Class.

   The New Object Class dialog box appears.

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

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

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

4. Click OK.

See Also:

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

Modifying Object Classes by Using Oracle Directory Manager

To modify an object class:

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

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

4. Click OK.

See Also:

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

Deleting Object Classes by Using Oracle Directory Manager

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

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

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

See Also: Example: Adding a New Attribute to an Auxiliary or User-Defined Object Class on page 6-15 for an example of adding attributes to an auxiliary object class

To delete an object class by using Oracle Directory Manager:

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

Managing Object Classes by Using Command-Line Tools

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

To add or modify schema components, use ldapmodify.

See: "ldapmodify Syntax" on page A-15

This section contains these examples:

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

Example: Adding a New Object Class

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

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

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

To load the file, enter this command:

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

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

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

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

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

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

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

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

About Attribute Management

This section contains these topics:

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

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

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

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

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

Rules for Adding Attributes

The rules for adding attributes are:

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

Rules for Modifying Attributes

The rules for modifying attributes are:

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

Rules for Deleting Attributes

The rules for deleting attributes are:

- You can delete only user-defined attributes. Do not delete attributes from the base schema.
- You can delete any attribute that is not referenced directly or indirectly by some other schema component.

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

Managing Attributes by Using Oracle Directory Manager

This section contains these topics:

See Also:

- "Attribute Options" on page 2-7 for information about attribute options
- "Managing Entries with Attribute Options by Using Oracle Directory Manager" on page 7-11 and "Managing Entries with Attribute Options by Using Command-Line Tools" on page 7-15 for instructions on adding and deleting attribute options and for searching for entries containing attribute options

Viewing All Directory Attributes by Using Oracle Directory Manager

To view attributes by using Oracle Directory Manager:

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

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

See Also: "Viewing Attributes for a Specific Entry by Using Oracle Directory Manager" on page 7-6 for instructions about how to view attributes for a specific entry

Searching for Attributes by Using Oracle Directory Manager

To search for attributes by using Oracle Directory Manager:

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

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

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

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

6. In the text box at the right end of the search criteria bar, type part or all of the value of the attribute for which you want to search. For example, to search for all attributes whose names begin with the letters `orcl`, you would type those letters in the text box at the right end of the search criteria bar and create the phrase `Name Begins With orcl`.
7. Beneath the Search Criteria field are five buttons described in the following table. Use these buttons to further refine your search.

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

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

Adding an Attribute by Using Oracle Directory Manager

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

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

Adding a New Attribute by Using Oracle Directory Manager

To add a new attribute:

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

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

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

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

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

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

Field	Description
Usage	Specify standards for how the attribute can be used. Options are: ■ `userApplications` Attributes whose values must be entered by the user, for example, `telephoneNumber` ■ `directoryOperation` Attributes whose values are entered by the directory server, for example, `creatorName` or `timeStamp` ■ `distributedOperation` ■ `dSAOperation` Attributes used for the internal operation of the server, for example, orclUpdateSchedule
Ordering	Specify standards for how precedence is established for values
Equality	Specify standards for how equality is determined in compare and search operations
Substring	Specify regular expression matching
Super	Add the super attribute for this attribute. To do this: 1. Click the Add button next to this field. The Super Attribute Selector appears. 2. Select the super attribute and click Select. 3. Repeat as needed. To delete a super attribute from the Super field, select it, then click Delete.

5. Click OK.

Note: To use this attribute, remember to declare it to be part of the attribute set for an object class. You do this by selecting Schema Management in the navigator pane, then, in the right pane, selecting the Object Classes tab page. For further instructions, see "Guidelines for Modifying Object Classes" on page 6-4.

Creating a New Attribute from an Existing One by Using Oracle Directory Manager

To add an attribute by copying an existing attribute:

1. In the navigator pane, select Schema Management.
2. In the right pane, select the Attributes tab.
3. In the Attributes tab page, select the attribute you want to copy.
4. Click the Create Like button at the bottom of the right pane. The New Attribute Type dialog box for that attribute appears. This dialog box contains two tab pages—General and Advanced—with fields in which you enter values either by typing or selecting from menus.
5. Select the General tab and enter values in each of the fields as described in the following table. You must always change the DN to that of the new attribute.

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

6. Select the Advanced tab and enter values in each of the fields as described in the following table.

Field	Description
Indexed	Select to add this attribute to the index, thereby making it available for use in a search. Only those attributes that have an equality matching rule can be indexed.
Usage	Specify standards for how the attribute can be used. Options are: ■ `userApplications` Attributes whose values must be entered by the user, for example, `telephoneNumber` ■ `directoryOperation` Attributes whose values are entered by the directory server, for example, `creatorName` or `timeStamp` ■ `distributedOperation` ■ `dSAOperation` Attributes used for the internal operation of the server, for example, orclUpdateSchedule
Ordering	Specify standards for how precedence is established for values
Equality	Specify standards for how equality is determined in compare and search operations
Substring	Specify regular expression matching
Super	Add the super attribute for this attribute. To do this: **1.** Click the Add button next to this field. The Super Attribute Selector appears. **2.** Select the super attribute and click Select. **3.** Repeat as needed. To delete a super attribute from the Super field, select it, then click Delete.

7. Click OK.

Modifying an Attribute by Using Oracle Directory Manager

To modify an attribute by using Oracle Directory Manager:

1. In the navigator pane, select Schema Management.
2. In the right pane, select the Attributes tab, then select an editable attribute in the list.
3. Click Edit. The Attribute dialog box displays two tab pages—General and Advanced—with fields in which you enter values either by typing or selecting from menus.
4. Select the General tab and enter values in each of the fields as described in the following table.

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

5. Select the Advanced tab and enter values in each of the fields as described in the following table.

Field	Description
Indexed	Select to add this attribute to the index, thereby making it available for use in a search. Only those attributes that have an equality matching rule can be indexed.
Usage	Specify standards for how the attribute can be used. Options are: ■ `userApplications` Attributes whose values must be entered by the user, for example, `telephoneNumber` ■ `directoryOperation` Attributes whose values are entered by the directory server, for example, `creatorName` or `timeStamp` ■ `distributedOperation` ■ `dSAOperation` Attributes used for the internal operation of the server, for example, orclUpdateSchedule
Ordering	Specify standards for how precedence is established for values
Equality	Specify standards for how equality is determined in compare and search operations
Substring	Specify regular expression matching
Super	Add the super attribute for this attribute. To do this: **1.** Click the Add button next to this field. The Super Attribute Selector appears. **2.** Select the super attribute and click Select. **3.** Repeat as needed. To delete a super attribute from the Super field, select it, then click Delete.

6. Click OK.

Deleting an Attribute by Using Oracle Directory Manager

Note: You can delete only user-defined attributes. Do not delete attributes from the base schema.

To delete an attribute:

1. In the navigator pane, select Schema Management.
2. In the right pane, select the Attributes tab, then select an editable attribute in the list.
3. Click Delete.

Indexing an Attribute by Using Oracle Directory Manager

Oracle Internet Directory uses indexes to make attributes available for searches. When Oracle Internet Directory is installed, certain attributes are already indexed. If you want to use additional attributes in search filters, you must index them.

Note: You can use Oracle Directory Manager to index an attribute only at the time when you create it. You cannot use Oracle Directory Manager to index an already existing attribute. To index an already existing attribute, use the Catalog Management tool as described in "Indexing an Attribute by Using Command-Line Tools" on page 6-31.

You can index only those attributes that have:

- An equality matching rule
- Matching rules supported by Oracle Internet Directory as listed in "Matching Rules" on page C-10
- No more than 28 characters in their names

Viewing Indexed Attributes by Using Oracle Directory Manager

To view indexed attributes:

1. In the navigator pane, select Schema Management.

2. In the right pane, select the Attributes tab. The Attributes tab displays all of the attributes in the schema. A selected check box in the Indexed column indicates an indexed attribute.

Adding an Index to an Attribute by Using Oracle Directory Manager

When you create an attribute as described in "Adding an Attribute by Using Oracle Directory Manager" on page 6-21, you use the New Attribute Type dialog box. On the Advanced tab page of that dialog box, you select the Indexed check box.

Dropping an Index from an Attribute by Using Oracle Directory Manager

To drop an index from an attribute:

1. In the navigator pane, select Schema Management.
2. In the right pane, select the Attributes tab.
3. Select the indexed attribute. Note that this must be an attribute that is editable as indicated by the icon to the left of the attribute name.
4. Click Drop Index.

Managing Attributes by Using Command-Line Tools

This section discusses adding, modifying, and indexing attributes by using command-line tools. This section contains these topics:

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

Adding and Modifying Attributes by Using ldapmodify

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 LDIF 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' )
```

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

See Also:

- "ldapmodify Syntax" on page A-15 for a detailed explanation of ldapmodify and its options
- "Viewing Syntaxes" on page 6-33 for instructions on how to view syntaxes by using either Oracle Directory Manager or ldapsearch

Deleting Attributes by Using ldapmodify

Note: You can delete only user-defined attributes. Do not delete attributes from the base schema.

To delete an attribute by using ldapmodify, type a command similar to the following at the system prompt:

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

The LDIF file contains data similar to this:

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

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

See Also:

- "ldapmodify Syntax" on page A-15 for a detailed explanation of ldapmodify and its options
- "Viewing Syntaxes" on page 6-33 for instructions on how to view syntaxes by using either Oracle Directory Manager or ldapsearch

Indexing an Attribute by Using Command-Line Tools

Oracle Internet Directory uses indexes to make attributes available for searches. When Oracle Internet Directory is installed, the entry `cn=catalogs` lists available attributes that can be used in a search.

If you want to use additional attributes in search filters, you must add them to the catalog entry. You can index only those attributes that have:

- An equality matching rule
- Matching rules supported by Oracle Internet Directory as listed in "Matching Rules" on page C-10
- No more than 28 characters in their names

You can index a new attribute—that is, one for which no data exists in the directory—by using ldapmodify. You can index an attribute for which data already exists in the directory by using the Catalog Management tool. You can drop an index from an attribute by using ldapmodify, but Oracle Corporation recommends that you use the Catalog Management tool.

Indexing an Attribute for Which *No* Data Exists by Using ldapmodify

Once you have defined a new attribute in the schema, you can add it to the catalog entry by using ldapmodify.

To add an attribute for which no directory data exists by using ldapmodify, import an LDIF file by using ldapmodify. For example, to add a new attribute `foo` that has already been defined in the schema, import the following LDIF file by using ldapmodify:

```
dn: cn=catalogs
changetype: modify
add: orclindexedattribute
orclindexedattribute: foo
```

You should not use this method to index an attribute for which data exists in the directory. To index such an attribute, use the Catalog Management tool.

Dropping an Index from an Attribute by Using ldapmodify

To drop an index from an attribute by using ldapmodify, specify `delete` in the LDIF file. For example:

```
dn: cn=catalogs
changetype: modify
```

```
delete: orclindexedattribute
orclindexedattribute: foo
```

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

Indexing an Attribute for Which Data Exists by Using the Catalog Management Tool

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

See Also: "Catalog Management Tool Syntax" on page A-40

Note: Be careful not to use the catalog.sh `-delete` option to remove indexes on attributes unless you are absolutely sure that the indexes were not created by the base schema that was installed with Oracle Internet Directory. Removing indexes from base schema attributes can adversely impact the operation of Oracle Internet Directory.

Viewing Matching Rules

This section contains these topics:

- Viewing Matching Rules by Using Oracle Directory Manager
- Viewing Matching Rules by Using ldapsearch

Note: Matching rules cannot be modified.

Viewing Matching Rules by Using Oracle Directory Manager

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server instance*, then select Schema Management.
2. In the right pane, select the Matching Rules tab. The fields in this tab page are shown as column heads. They are:

Column Head	Meaning
Name	Name of the attribute matching rule

Column Head	Meaning
Object ID	Unique identifier of this matching rule
Description	Words describing the matching rule (optional)
Syntax	Syntax used with this matching rule

Viewing Matching Rules by Using ldapsearch

Use ldapsearch on the subentry `cn=subSchemaSubentry`.

See Also: "ldapsearch Syntax" on page A-22

Viewing Syntaxes

This section contains these topics:

- Viewing Syntaxes by Using Oracle Directory Manager
- Viewing Syntaxes by Using by Using ldapsearch

Note: Syntaxes cannot be modified.

Viewing Syntaxes by Using Oracle Directory Manager

To view syntaxes by using Oracle Directory Manager:

1. In the navigator pane, select Schema Management.
2. In the right pane, select the Syntaxes tab. The fields in this tab page are shown as column heads. They are:
   - Description—Name of the attribute syntax
   - Object ID—Unique identifier of this syntax

Viewing Syntaxes by Using by Using ldapsearch

Use ldapsearch on the subentry `cn=subSchemaSubentry`.

See Also: "ldapsearch Syntax" on page A-22

7

Managing Directory Entries

This chapter explains how to view, add, modify, and delete entries.

This chapter contains these topics:

- Managing Entries by Using Oracle Directory Manager
- Managing Entries by Using Command-Line Tools
- Managing Entries by Using Bulk Tools
- Managing Knowledge References and Referrals

> **See Also:** Chapter 2, "Concepts and Architecture" for an overview of directory entries, directory information trees, distinguished names, and relative distinguished names

Managing Entries by Using Oracle Directory Manager

This section contains these topics:

Searching for Entries by Using Oracle Directory Manager

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

To display an entry, in the navigator pane, expand Oracle Internet Directory Servers > *directory server instance* > Entry Management to display its subtree.

The root of the tree is listed first, then the second level, and so forth, moving from left to right. The subtree lists the **RDN** of each entry in hierarchical order. To see the lower level entries within any subtree, click the plus sign (+) to the left of the parent entry.

To search for a directory entry:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory server instance*, and select Entry Management. The Search fields appear in the right pane.
2. In the Root of the Search field, enter the **DN** of the root of your search.

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

   ```
   ou=Manufacturing,ou=Americas,o=IMC,c=US
   ```

   You would therefore type that DN in the Root of the Search text box.

You can also select the root of your search by browsing the **directory information tree (DIT)**. To do this:

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

b. Click the plus sign (+) next to tree view to display its entries.

c. Continue navigating to the entry that represents the level you want for the root of your search.

d. Select that entry, then click OK. The DN for the root of your search appears in the Root of the Search text box in the right pane.

3. In the Max Results (entries) box, type the maximum number of entries you want your search to retrieve. The default is 200. The directory server retrieves the value you set, up to 1000.

4. In the Max Search Time (seconds) box, type the maximum number of seconds for the duration of your search. The value you enter here must be at least that of the default, namely, 25. The directory server searches for the amount of time you specify, up to one hour.

5. In the Search Depth list, select the level in the DIT to which you want to search.

   The options are:

   - Base: Retrieves a particular directory entry. Along with this search depth, you use the Search criteria bar to select the attribute `objectClass` and the filter `Present`.
   - One Level: Limits your search to all entries beginning one level down from the root of your search
   - Subtree: Searches entries within the entire subtree, including the root of your search

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

   a. From the list at the left end of the search criteria bar, select an attribute of the entry for which you want to search. Because not all attributes are used in every entry, be sure that the attribute you specify actually corresponds to one in the entry for which you are looking. Otherwise, the search will fail.

b. From the list in the middle of the search criteria bar, select a filter. Options are:

Filter	Description
Begins With	Searches by using only the first few characters of the attribute's value. For example, `cn Begins With Fran` retrieves all entries in which the first few letters of the `cn` attribute are `Fran`. These would include Frank, Fran, Frances, Franklin, etc.
Ends With	Searches for an entry by using only the last few characters of the specified attribute's value. For example, `cn Ends With son` retrieves Baldisson, Jacobson, Johnson, etc.
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter. For example, `cn Contains Wins` retrieves all entries in which the `cn` attribute contains the letters `wins`. These would include Winslow, Czerwinski, Winship, etc.
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter. For example, `cn Exactly Matches Franklin Baldwins` retrieves all entries in which the `cn` attribute has the value `Franklin Baldwins`.
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. For example, `cn Greater or Equal Frank` retrieves all entries with `cn` attributes that range from the first Frank to the end of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. For example, `cn Less or Equal Frank` retrieves all `cn` attributes from the first Frank to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. The phrase `cn Present` retrieves all entries with the `cn` attribute at that level of the tree.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you just selected. For example, if the attribute you selected was `cn`, you could type the particular common name you want to find.

7. To further refine your search, use the buttons in the Search Criteria box to enhance the search criteria bar.

Button	Description
New	Creates a new search criteria bar in the Search Criteria field. This button is enabled only when the Search Criteria field is empty.
And	Creates another search criteria bar in the Search Criteria field. Matches all entries with one specified attribute with those that also have another specified attribute. For example, `cn=Baldwins And title=Laborer` retrieves all Baldwins who are also laborers.
Or	Creates another search criteria bar in the Search Criteria field. Matches all entries with either one specified attribute or another. For example, `title=Laborer Or title=Foreman` retrieves all employees who are either laborers or foremen.
Not	Negates the criterion in the selected search criteria bar and retrieves all entries that do not have the specified criterion. For example, `cn=Frank And Not title=Laborer` retrieves all persons named Frank who are not laborers.
Delete	Deletes a selected search criteria bar
Advanced	Adds a search criteria bar when including attribute options in the search. Use this syntax: *`attribute;attribute_option filter attribute_option_value`* For example, cn;lang_sp=J* retrieves all attribute option values for `cn;lang_sp=` that begin with the letter J. **Note:** Before an attribute option can be used in searches, the parent attribute of that attribute option must be indexed. For example, in the case of the attribute option `carLicense;lang_sp`, the `carLicense` attribute must be indexed before the `carLicense;lang_sp` attribute option can be used in searches. **See Also:** ■ "Indexing an Attribute by Using Oracle Directory Manager" on page 6-28 ■ "Indexing an Attribute by Using Command-Line Tools" on page 6-31

8. Click Search. The results of your search appear in the Distinguished Name box.

See Also: "Configuring Searches" on page 5-20 for instructions on setting the number of entries to display in searches, and to set the time limit for searches

Viewing Attributes for a Specific Entry by Using Oracle Directory Manager

Once you have displayed the results of your search, click the entry whose attributes you want to view. An Entry dialog box displays the attributes for that entry.

Some attributes can also be DNs. For example, one attribute for a given employee might be that employee's manager who, in turn, has a DN. In this case, when you display the Entry dialog box for the employee, you would see a Browse button next to the Manager text box. To find information about that manager, click Browse to display the Directory: Entry Management dialog box, then follow the steps mentioned in "Searching for Entries by Using Oracle Directory Manager" on page 7-2.

> **See Also:** "Viewing All Directory Attributes by Using Oracle Directory Manager" on page 6-18 for instructions about how to view all attributes in the directory

Adding Entries by Using Oracle Directory Manager

This section tells you how to add entries for individuals and groups.

> **Note:** When you add or modify an entry, the Oracle directory server does not verify the syntax of the attribute values in the entry.

Adding a New Entry by Using Oracle Directory Manager

To add or delete entries with Oracle Directory Manager, you must have write access to the parent entry and you must know the DN for the new entry.

To add a new entry:

1. Expand Oracle Internet Directory Servers > *directory_server_instance*, then select Entry Management.
2. On the toolbar, click Create. The New Entry dialog box appears.
3. In the Distinguished Name field, type the full DN. You may also click Browse to locate and select the DN of the parent for the entry you want to add. The entry you select appears in the Distinguished Name field. To the left of that parent DN, type the RDN for your new entry, followed by a comma.
4. To specify the **object classes** for the new entry, next to the Object Classes box, click Add. The Super Class Selector dialog box appears.

5. In the Super Class Selector dialog box, select an object class, then click Select. As you select from the object class list, mandatory and optional attributes populate the windows in the tab pages in the lower half of the New Entry dialog box. You must enter values into the mandatory attributes fields. You are not required to enter values into the optional attributes fields.
6. When you have selected the object classes and provided values for the appropriate attributes, click OK.

Adding an Entry by Copying an Existing Entry in Oracle Directory Manager

You can use Oracle Directory Manager to create a new entry by copying from an existing entry and changing its DN. When you do this, you should also change the attributes, such as name and address, so that they correspond to the new DN. To add an entry, you must have write access to its parent.

> **Tip:** You can find a template for the new DN by looking up other similar entries in the search pane.

To add an entry by copying an existing entry:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*, then select Entry Management. in the right pane, the Search interface appears. Use it to search for an entry that you want to use as a template.
2. From the entries retrieved, double-click one that you want to use as your template. The Entry dialog box for that entry appears.
3. In the Entry dialog box, click Create Like. A New Entry: Create Like dialog box appears.
4. Change critical fields to tailor this entry to the one that you want to create. You must always change the DN and the common name in this operation, or the pane will not save your new entry data. For example, if you create an entry for Henri Latrobe by using the entry for Henri Latour as the template, then you have to change `cn=Henri Latour` in the DN to `cn=Henri Latrobe`. You also must change any other attributes that must be unique, such as employee number and telephone number.
5. Click OK to save your changes.

> **See Also:** The online help for this dialog box for details about adding information into fields

Example: Adding a User Entry by Using Oracle Directory Manager

In this example, we create a user named Anne Smith and assign her a password.

1. Login as the administrator.
2. Expand Oracle Internet Directory Services > directory_*server_instance*, and select Entry Management.
3. On the toolbar, click the Create button. The New Entry dialog box appears.
4. In the Distinguished Name field, type the full DN. You may also click the Browse button to locate the DN of the parent for this entry, then type the RDN, namely, `cn=Anne Smith`, followed by a comma, to the left of that parent DN.
5. To the right of the Object Classes box, Click Add. The Super Class Selector dialog box appears.
6. In the Super Class Selector dialog box, select the `person` object class, then click Select. This returns you to the New Entry dialog box.
7. In the New Entry dialog box, click the Optional Properties tab, and scroll to the `userPassword` window.
8. Type the password for Anne Smith.

Adding Group Entries by Using Oracle Directory Manager

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

You determine membership in the group by adding DNs to the multivalued attribute `member` if the entry belongs to the `groupOfNames` object class, or `uniqueMember` if the entry belongs to the `groupOfUniqueNames` object class.

To add a group entry:

1. Expand Oracle Internet Directory Servers > *directory_server_instance*, then select Entry Management.
2. On the toolbar, click Create. The New Entry dialog box appears.
3. In the Distinguished Name field, type the full DN. You may also use the Browse button to locate the DN of the parent for the entry you want to add, then type the RDN for the new entry, followed by a comma, to the left of that parent DN.
4. To specify the object classes you want to use for the new entry, to the right of the Object Classes box, click Add. The Super Class Selector dialog box appears.

5. In the Super Class Selector dialog box, select the `top` object class, then click the Select button. The `top` object class appears in the Object Classes box of the New Entry dialog box.
6. In the same way:
   a. To the right of the Object Classes box, click Add.
   b. From the Super Class Selector dialog box, select the `groupOfNames` or `groupOfUniqueNames` object class.
   c. Click Select. The object class you selected appears in the Object Classes window of the New Entry dialog box.
7. Enter the mandatory and optional attributes for your group entry.

   If you selected the `groupOfNames` object class, a Browse button appears next to some of the fields, for example, the member field on the Mandatory Properties tab page. To enter a mandatory property by browsing:
   a. Click Browse. The Directory: Entry Management dialog box appears.
   b. Use this dialog box to search for a particular entry you want to add to the list.
   c. In the Distinguished Name window of the Directory: Entry Management dialog box, select the entry, then click OK. This returns you to the New Entry dialog box. The entry you just selected is added to the list in the members window.
8. Click OK.

See Also:

- "Searching for Entries by Using Oracle Directory Manager" on page 7-2 for instructions on using the search pane
- "Access Control Groups" on page 13-3 for instructions on setting access control policies for group entries
- Globalization Support on page 2-14 and Chapter 13, "Directory Access Control" for information about access privileges

Modifying Entries by Using Oracle Directory Manager

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

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

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

- You cannot add mandatory attributes to an object class already in use by some entries. You may add optional attributes to object classes that are already in use by entries. If you add optional attributes to an object class already in use by some entries, no special rules apply—they are added as empty attributes to those entries.

> **Note:** When you add or modify an entry, the Oracle directory server does not verify the syntax of the attribute values in the entry.

To modify an entry:

1. Perform a search for the entry you want to modify as described in "Searching for Entries by Using Oracle Directory Manager" on page 7-2.
2. In the Distinguished Name box of the right pane, select the entry you want to modify.
3. Click Edit. The Entry dialog box appears.
4. Click OK.

Example: Modifying a User Entry by Using Oracle Directory Manager

In this example, we modify the password for the entry we created for Anne Smith in the section "Example: Adding a User Entry by Using Oracle Directory Manager" on page 7-8.

1. Perform a search for the Anne Smith entry.
2. In the right pane, in the Distinguished Name box, select the entry for Anne Smith.
3. Click Edit.

4. In the Entry dialog box, scroll to the `userPassword` window and modify the value.
5. Click OK.

Managing Entries with Attribute Options by Using Oracle Directory Manager

This section tells you how to add, modify, and delete attribute options.

> **See Also:** "Searching for Entries by Using Oracle Directory Manager" on page 7-2 for instructions on searching for entries with attribute options

Adding an Attribute Option to an Existing Entry by Using Oracle Directory Manager

> **Note:** In Oracle Internet Directory Release 9.0.2, Oracle Directory Manager does not allow you to add an attribute option to an entry when you create the entry. You can use Oracle Directory Manager to add attribute options only to already existing entries.

To add an attribute option to an existing entry:

1. Expand Oracle Internet Directory Servers > *directory server instance* > Entry Management, then select the entry to which you want to add an attribute option. The corresponding tab pages appear in the right pane.
2. In the right pane, in the Properties tab page, in the View Properties field, select Advanced. The Properties tab page changes accordingly.
3. In the Attribute field, select the attribute to which you want to add the option, for example, `ou`.
4. In the Attribute Options field, enter the attribute option, for example, `lang-en`.
5. In the Attribute Value field, enter the value of the attribute option you just specified, for example, `Server Technologies`. To add more than one attribute value for the specified attribute option, separate the values by using a semicolon.
6. Click Apply.

Modifying an Attribute Option by Using Oracle Directory Manager

To modify an attribute option:

1. Expand Oracle Internet Directory Servers > *directory server instance* > Entry Management, then select the entry from which you want to delete an attribute option. The corresponding tab pages appear in the right pane.
2. In the Properties tab page, in the View Properties field, select either Only Non-null Values or All.
3. Scroll to the field containing the attribute option you want to modify.
4. Modify the value in the field.
5. Click Apply.

Deleting an Attribute Option by Using Oracle Directory Manager

To delete an attribute option:

1. Expand Oracle Internet Directory Servers > *directory server instance* > Entry Management, then select the entry from which you want to delete an attribute option. The corresponding tab pages appear in the right pane.
2. In the Properties tab page, in the View Properties field, select either Only Non-null Values or All.
3. Scroll to the field containing the attribute option you want to delete.
4. Delete the value in the field.
5. Click Apply.

Managing Entries by Using Command-Line Tools

This section points you to the command-line tools you can use in managing entries. It also provides several examples of entry management by using command-line tools. It contains these topics:

Command-Line Tools for Managing Entries

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

Tool	Task(s)	Syntax and Usage Notes
ldapsearch	Search for directory entries.	"ldapsearch Syntax" on page A-22
ldapbind	Authenticate a user or client to a directory server. Verify that you can connect a client to a server.	"ldapbind Syntax" on page A-8
ldapadd	Add entries one at a time. Add new configuration set entries. Configure a server with an input file.	"ldapadd Syntax" on page A-4
ldapaddmt	Add several entries concurrently by using this multithreaded tool.	"ldapaddmt Syntax" on page A-6
ldapmodify	Create, update, and delete attribute data for an entry. Modify configuration set entries. Modify DN or RDN of an entry.	"ldapmodify Syntax" on page A-15
ldapmodifymt	Modify several entries concurrently by using this multithreaded tool.	"ldapmodifymt Syntax" on page A-20
ldapdelete	Delete entries.	"ldapdelete Syntax" on page A-11
ldapcompare	Compare attribute values you specify with those in a directory entry.	"ldapcompare Syntax" on page A-9

Tool	Task(s)	Syntax and Usage Notes
ldapmoddn	Modify the DN or RDN of an entry. Rename an entry or a subtree. Move an entry or a subtree under a new parent.	"ldapmoddn Syntax" on page A-13

Example: Adding a User Entry by Using ldapadd

The following example shows an LDIF file, named `entry.ldif`, for the user entry for an employee named John:

```
dn: cn=john, c=us
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: john
cn;lang-fr:Jean
cn;lang-en-us:John
sn: Doe
jpegPhoto: /photo/john.jpg
userpassword: welcome
```

This file contains the `cn`, `sn`, `jpegPhoto`, and `userpassword` attributes.

For the `cn` attribute, it specifies two options: `cn;lang-fr`, and `cn;lang-en-us`. These options return the common name in either French or American English.

For the `jpegPhoto` attribute, it specifies the path and file name of the corresponding JPEG image you want to include as an entry attribute.

Note: When you add or modify an entry, the Oracle directory server does not verify the syntax of the attribute values in the entry.

Example: Modifying a User Entry by Using ldapmodify

The following example changes the password for a user named Audrey from `welcome` to `audreyspassword`. As in the example above, the data for this user entry is in the `entry.ldif` file. This file contains the following:

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

Issue this command to modify the file:

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

where -v specifies verbose mode.

Note: When you add or modify an entry, the Oracle directory server does not verify the syntax of the attribute values in the entry.

Managing Entries with Attribute Options by Using Command-Line Tools

This section provides examples of how to add and delete attribute options, and how to search for entries with attribute options.

Example: Adding an Attribute Option by Using ldapmodify

the data for this user entry is in the `entry.ldif` file. This file contains the following:
Issue this command to modify the file:

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

Example: Deleting an Attribute Option by Using ldapmodify

Issue this command to modify the file:

Example: Searching for Entries with Attribute Options by Using ldapsearch

The following example retrieves entries with common name (cn) attributes that have an option specifying a language code attribute option. This particular example retrieves entries in which the common names are in French and begin with the letter R.

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub "cn;lang-fr=R*"
```

Suppose that, in the entry for John, no value is set for the `cn;lang-it` language code attribute option. In this case, the following example fails:

```
ldapsearch -p 389 -h myhost -b "c=us" -s sub "cn;lang-it=Giovanni
```

See Also: "Attribute Options" on page 2-7

Managing Entries by Using Bulk Tools

This section lists and describes some of the more common tasks you perform with bulk tools.

This section contains these topics:

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

Note: If you do not use the bulkload utility to populate the directory, then you must run the oidstats.sh tool to avoid significant search performance degradation.

See Also:

- "OID Database Statistics Collection Tool Syntax" on page A-55 for a description and syntax for the oidstats.sh tool
- "Using Bulk Tools" on page 4-12 for an overview of these tools

Importing an LDIF File by Using bulkload

To import an LDIF file, you use the bulkload utility. This section discusses the tasks to process an LDIF file through bulkload.

Note: The bulkload utility expects an empty directory and will either fail or overwrite if there are existing entries.

Before performing a bulk load, stop the Oracle Internet Directory processes. See Chapter 3, "Preliminary Tasks and Information" for instructions on stopping directory server instances.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

This section contains these topics:

- Task 1: Back Up the Oracle Server
- Task 2: Find Out the Oracle Internet Directory Password
- Task 3: Check Input for Schema and Data Consistency Violations
- Task 4: Generate the Input Files for SQL*Loader
- Task 5: Load the Input Files
- If Bulk Loading Fails

Task 1: Back Up the Oracle Server

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

See Also: *Oracle9i User-Managed Backup and Recovery Guide* in the Oracle Database Documentation Library

Task 2: Find Out the Oracle Internet Directory Password

To use bulkload and the other shell script tools that have commands that end with `.sh`, you must provide the Oracle Internet Directory password. The default password is `ods`, although the system administrator can change it by using the **OID Database Password Utility**.

> **See Also:** "Using the OID Database Password Utility" on page 4-14

Task 3: Check Input for Schema and Data Consistency Violations

On Solaris, the `bulkload.sh` file usually resides in $*ORACLE_HOME*`/ldap/bin`. On Windows NT, this file usually resides in *ORACLE_HOME*`\ldap\bin`.

Check the input file by typing:

```
bulkload.sh -connect net_service_name -check path_to_ldif-filename
```

All schema violations are reported in $*ORACLE_HOME*`/ldap/log/schemacheck.log`

If any violations are detected in the input file, use an ASCII text file editor to fix or remove them. If there are any duplicate entries, their DNs are logged in $*ORACLE_HOME*`/ldap/log/duplicate.log`.

Task 4: Generate the Input Files for SQL*Loader

After you have fixed any errors in the input file, rerun bulkload with the `-generate` option as shown in the following example. During this step, LDIF data is converted to SQL*Loader specific format.

```
bulkload.sh -connect net_service_name -generate ldif-filename
```

All loading errors are reported in $*ORACLE_HOME*`/ldap/log`

When this command completes successfully, it generates `*.dat` files in the `$ORACLE_HOME/ldap/load` directory to be used by SQL*Loader in `-load` mode. Do not modify these files.

Task 5: Load the Input Files

After you have generated the input files, rerun bulkload with the `-load` option. During this step, the `*.dat` files, which are in Oracle SQL*Loader specific format, are loaded into the database and the attribute indexes are created. The syntax is:

```
bulkload.sh -connect net_service_name -load
```

If Bulk Loading Fails

All loading errors are reported in the $*ORACLE_HOME*`/ldap/log/directory` with the file extension `.bad`.

If bulk loading fails, the database could be left in an inconsistent state. It may be necessary to restore the database to its state prior to the bulk loading operation.

Converting Directory Data to LDIF

Converting directory data to LDIF by using LDIF Writer makes the data available for loading into a new node in a replicated directory or into another node for backup storage.

See Also: "ldifwrite Syntax" on page A-39

Modifying a Large Number of Entries

The bulkmodify utility enables you to modify a large number of existing entries efficiently.

See Also: "bulkmodify Syntax" on page A-37

Deleting a Large Number of Entries

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

See Also: "bulkdelete Syntax" on page A-34

Managing Knowledge References and Referrals

A **knowledge reference**, also called a **referral**, is represented in the directory as a particular type of **entry**. When you create a knowledge reference entry, you associate it with the `referral` **object class** the and `extensibleObject` object class. Typically, you create knowledge reference entries at the place in the **DIT** where you want to establish the partition.

A knowledge reference provides users with a referral containing an LDAP URL. You enter these URLs as values for the `ref` attribute. There can be multiple `ref` attributes specified for any knowledge reference entry. Similarly, there can be multiple knowledge reference entries in the DIT.

> **See Also:** "Partitioning" on page 2-25 for an overview of knowledge references and a description of **smart knowledge references** and **default knowledge references**

This section contains these topics:

- Configuring Smart Referrals
- Configuring Default Referrals

Configuring Smart Referrals

A search result can contain regular entries along with knowledge references. When a user performs a search operation, Oracle Internet Directory looks for the knowledge reference entry within the specified scope of the search. If it finds the knowledge reference, then Oracle Internet Directory returns a referral to the client.

If a user performs an add, delete, or modify operation on an entry located below the knowledge reference entry, then Oracle Internet Directory returns the referral.

For example, suppose you want to partition the DIT based on the geographical location of the directory servers. In this example, assume that:

- The `c=us` naming context is held locally on Server A and Server B in the United States.
- The `c=uk` naming context is held locally on Server C and Server D in the United Kingdom.

In this case, you would configure knowledge references between these two naming contexts as follows:

1. On Server A in the United States, configure a knowledge reference for the `c=uk` object on Server C and Server D:

```
dn: c=uk
c: uk
ref: ldap://host C:389/c=uk
ref: ldap://host D:686/c=uk
objectclass: top
objectclass: referral
```

```
objectClass: extensibleObject
```

2. Configure a similar knowledge reference on Server C in the United Kingdom for the `c=us` object on Server A and Server B:

```
dn: c=us
c: us
ref: ldap://host A:4000/c=us
ref: ldap://host B:5000/c=us
objectclass: top
objectclass: referral
objectClass: extensibleObject
```

Results:

- A client querying Server A with base `o=foo,c=uk` receives a referral.
- A client querying Server C with base `o=foo,c=us` receives a referral.
- An add operation of `o=foo,c=uk` on either Server A or Server B fails. Instead, Oracle Internet Directory returns a referral.

Configuring Default Referrals

Oracle Internet Directory uses the `namingcontext` attribute in the **DSE** to determine all the **naming contexts** held locally by the server. Be sure that the `namingContext` attribute correctly reflects the naming context information.

You specify default referrals by entering a value for the `ref` attribute in the DSE entry. If the `ref` attribute is not in the DSE entry, then no default referral is returned.

When configuring a default referral, do not specify the DN in the LDAP URL.

For example, suppose that the DSE entry on Server A contains the following `namingContext` value:

```
namingcontext: c=us
```

Further, suppose that the default referral is:

```
Ref: ldap://host PQR:389
```

Now, suppose that a user enters an operation on Server A that has a base DN in the naming context `c=canada`, for example:

```
ou=marketing,o=foo,c=canada
```

This user would receive a referral to the host PQR. This is because Server A does not hold the `c=canada` base DN, and the `namingcontext` attribute in its DSE does not hold the value `c=canada`.

> **See Also:** "About Knowledge References and Referrals" on page 2-26 for a conceptual discussion of knowledge references

8

Globalization Support in the Directory

Oracle Internet Directory uses Globalization Support to store, process and retrieve data in native languages. It ensures that Oracle Internet Directory utilities and error messages automatically adapt to the native language and locale.

This chapter discusses Globalization Support as used by Oracle Internet Directory and tells you the required NLS_LANG environment variables for the various components and tools in an Oracle Internet Directory environment.

> **See Also:** "Globalization Support" on page 2-14 prior to configuring Globalization Support

This chapter contains these topics:

The NLS_LANG Environment Variable

The NLS_LANG parameter has three components—`language`, `territory`, and `charset`—in the form:

`NLS_LANG = language_territory.charset`

Each component controls the operation of a subset of Globalization Support features.

Component	Description
language	Specifies conventions such as the language used for Oracle messages, day names, and month names. Each supported language has a unique name—for example, American English, French, or German. The language argument specifies default values for the territory and character set arguments, so either (or both) `territory` or `charset` can be omitted. If language is not specified, the value defaults to American English. **See Also:** *Oracle9i Database Globalization Support Guide* in the Oracle Database Documentation Library for a complete list of languages
territory	Specifies conventions such as the default calendar, collation, date, monetary, and numeric formats. Each supported territory has a unique name; for example, America, France, or Canada. If territory is not specified, the value defaults to America. **See Also:** *Oracle9i Database Globalization Support Guide* in the Oracle Database Documentation Library for a complete list of territories
charset	Specifies the character set used by the client application (normally that of the user's terminal). Each supported character set has a unique acronym, for example, US7ASCII, WE8ISO8859P1, WE8DEC, WE8EBCDIC500, or JA16EUC. Each language has a default character set associated with it. Default values for the languages available on your system are listed in your operating system installation guide or administrator's guide. **See Also:** *Oracle9i Database Globalization Support Guide* in the Oracle Database Documentation Library for a complete list of character sets

Note: All components of the NLS_LANG definition are optional, that is, any item left out will default.

Also, if you specify `territory` or `charset`, you *must* include the preceding delimiter [underscore (_) for `territory`, and period (.) for `charset`], otherwise the entire value will be parsed as a language name.

You can set NLS_LANG as an environment variable at the command line. The following are examples of legal values for NLS_LANG:

- `AMERICAN_AMERICA.UTF8`
- `JAPANESE_JAPAN.UTF8`

Using Non-UTF-8 Databases

You can run the Oracle directory server and database tools on a non-UTF-8 database, but be sure that the client and database character sets are the same. Otherwise, you can lose data during ldapadd, ldapdelete, ldapmodify, or ldapmodifydn operations. For example, suppose that you perform an ldapadd operation using a multibyte character set on an underlying database that uses only single-byte characters. You will lose data because not all of the bytes you enter will be accepted by the database.

Using Globalization Support with LDIF Files

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

Attribute types are always ASCII strings that cannot contain multibyte characters. Oracle Internet Directory does not support multibyte characters in attribute type names. However, Oracle Internet Directory does support attribute *values* containing multibyte characters such as those in the simplified Chinese (.ZHS16GBK) character set.

Attribute values can be encoded in different ways to allow Oracle Internet Directory tools to interpret them properly. There are two scenarios:

- An LDIF file Containing Only ASCII Strings

An LDIF file Containing Only ASCII Strings

In this scenario, character strings for attribute values are also in ASCII.

Because all tools use the UTF-8 character set by default, and ASCII is a proper subset of UTF-8, all tools can interpret these files. The same is true of keyboard input of values that are simply ASCII strings.An LDIF file Containing UTF-8 Encoded Strings

An LDIF file Containing UTF-8 Encoded Strings

In this scenario, character strings for attribute values are also in UTF-8.

Because all tools use the UTF-8 character set by default, all tools can interpret these files. The same is true of keyboard input of values which are UTF-8 strings.

In such a file, some characters may be multibyte. Multibyte characters strings can be present in the LDIF files as attribute values or given as keyboard input. They can be encoded in their native character set or in UTF-8. They can also be BASE64 encoded representations of either the native or the UTF-8 string.

Consider the following cases:

- CASE 1: Native Strings (Non-UTF-8)
- CASE 2: UTF-8 Strings
- CASE 3: BASE64 Encoded UTF-8 Strings
- CASE 4: BASE64 Encoded Native Strings

Because the directory server understands and expects only UTF-8 encoded strings, cases 1, 3, and 4 need to undergo conversion to UTF-8 strings before they can be sent to the LDAP server.

CASE 1: Native Strings (Non-UTF-8)

Use the `-E` argument in the command-line tools, ldifwrite, and bulkmodify. Use the `-encode` argument in the bulkload and bulkdelete tools.

This example converts simplified Chinese native strings to UTF-8. The baseDN can be a simplified Chinese string:

```
ldapsearch -h my_host -p 389 -E ".ZHS16GBK" -b base_DN  -s base  "objectclass=*"
```

CASE 2: UTF-8 Strings

No conversion is required.

CASE 3: BASE64 Encoded UTF-8 Strings

You need to use neither the `-E` argument in the command-line tools, ldifwrite, and bulkmodify, nor the `-encode` argument in bulkload and bulkdelete. Oracle Internet Directory tools automatically decode BASE64 encoded UTF-8 strings to UTF-8 strings.

CASE 4: BASE64 Encoded Native Strings

Use the `-E` argument in the command-line tools, ldifwrite, and bulkmodify. Use the `-encode` argument in the bulkload and bulkdelete tools.

Oracle Internet Directory tools automatically decode BASE64 encoded native strings to simple native strings. The native strings are then converted to the equivalent UTF-8 strings.

Note: In any given input file, only one language set may be used.

Using Globalization Support with Command-Line Tools

The Oracle Internet Directory command-line tools read keyboard input or LDIF file input in the following ways:

- ASCII characters only
- Non-ASCII input (native language character set)
- BASE64 encoded values of UTF-8 or native strings (from LDIF file only)

If the character set being given as input from an LDIF file or keyboard is not UTF-8, then the command-line tools need to convert the input into UTF-8 format before sending it to the LDAP server.

You enable the command-line tools to convert the input into UTF-8 by specifying the `-E` argument when using each tool.

This section contains these topics:

Specifying the -E Argument When Using Each Tool

The client tools always assume UTF-8 to be the character set unless otherwise specified by the `-E` argument. The BASE64-encoded values are decoded, and then the decoded buffer is converted to UTF-8 if the `-E` argument is specified. For example, if you specify `-E ".ZHS16GBK"`, then the decoded buffer is converted from simplified Chinese to UTF-8 before being sent to the LDAP server.

Specifying the `-E` argument ensures that proper character set conversion can occur from the character set you specify for the `-E` argument (`-E ".`*`character_set`*`"`) to the.UTF-8 character set.

The command-line tools use the `-E` argument to process the input in the character set specified for the `-E` argument. They display their output in the character set specified in the NLS_LANG environment variable.

For example, to add entries from an LDIF file encoded in the simplified Chinese character set (.ZHS16GBK) by using ldapadd, type:

```
ldapadd -h myhost -p 389 -E ".ZHS16GBK" -f my_ldif_file
```

In this example, the ldapadd tool converts the characters from `".ZHS16GBK"` (simplified Chinese character set) to `".UTF8"` (UTF-8 character set) before they are sent across the wire to the LDAP server.

Examples: Using the -E Argument with Command-Line Tools

The following table provides additional examples of how to use the `-E` argument correctly for each command-line tool. In each example, the command converts data from simplified Chinese, as specified by the value `".ZHS16GBK"`, to UTF-8. For example, in each command, the values for the `-D` and `-w` options are in simplified Chinese. Specifying the `-E` argument converts them to UTF-8.

Note that, in the examples in the following table, we do not show any actual characters belonging to `.ZHS16GBK` character set. These examples would, therefore, work without the `-E` argument. However, if the argument values contained actual characters in the `.ZHS16GBK` character set, then we would need to use the `-E` argument.

> **See Also:** Appendix A, "Syntax for LDIF and Command-Line Tools" for syntax and usage notes for each of the command-line tools

Tool	Example
ldapbind	`ldapbind -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password`
ldapsearch	`ldapsearch -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password`
ldapadd	`ldapadd -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password`
ldapaddmt	`ldapaddmt -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password`
ldapmodify	`ldapmodify -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password`
ldapmodifymt	`ldapmodifymt -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password`
ldapdelete	`ldapdelete -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password`
ldapcompare	`ldapcompare -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password -b "ou=Construction,ou=Manufacturing,o=acme,c=us" -a title -v manager`
ldapmoddn	`ldapmoddn -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password -b "cn=Franklin Badlwins,ou=Construction,ou=Manufacturing,c=us,o=acme" -N "ou=Contracting,ou=Manufacturing,o=acme,c=us" -r`

Setting NLS_LANG in the Client Environment

If the output required by the client is UTF-8, then you do not need to set the NLS_LANG environment variable. In this case, the NLS_LANG environment variable defaults to `.UTF8`, and both the input path from client to server, and the output path from server to client, do not require any character set conversion.

If the output required by the client is *not* UTF-8, then you must set the NLS_LANG environment variable. This ensures that proper character set conversion can occur from the UTF-8 character set to the character set required by the client.

For example, if the NLS_LANG environment variable is set to the simplified Chinese character set, then the command-line tool displays output in that character set. Otherwise the output defaults to the UTF-8 character set.

Note: If you are using Windows NT, then, to use the command-line tools after server startup, you must reset NLS_LANG in an MS-DOS window. Set it to the character set that matches the code page of your MS-DOS session. UTF-8 cannot be used.See the *Oracle9i Database installation guide for Windows*in the Oracle Database Documentation Library for more information on which character set to use for command-line tools in an MS-DOS session.

If you are using a pre-installed Oracle9*i* release 9.0.1 database with Oracle Internet Directory, then you must also set the database character set to UTF-8. See the *Oracle9i Database Globalization Support Guide*in the Oracle Database Documentation Library and *Oracle9i Database installation guide for Windows* for more information.

Be careful not to change the NLS_LANG parameter value in the registry.

Using Globalization Support with Bulk Tools

Oracle Internet Directory ensures that the reading and writing of text data from and to LDIF files are done in UTF-8 encoding as specified by the LDAP standard.

This section provides an example of the argument you use for each of the following bulk tools:

- Using Globalization Support with bulkload
- Using Globalization Support with ldifwrite
- Using Globalization Support with bulkdelete
- Using Globalization Support with bulkmodify

See Also: "Bulk Tools Syntax" for a list of arguments for each bulk tool

Using Globalization Support with bulkload

Add to the command the argument `-encode "`*`character_set`*`"` where the input LDIF file is encoded in *`"character_set"`*.

For example:

```
bulkload.sh -connect net_service_name -encode ".ZHS16GBK" my_ldif_file
```

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Using Globalization Support with ldifwrite

The ldifwrite utility always writes BASE64 encoded values for multibyte strings.

The BASE64 encoding could be of the UTF-8 strings as they are stored in the directory server, or of native strings as specified by the NLS_LANG environment variable setting when running ldifwrite.

For example:

```
ldifwrite -c net_service_name -b baseDN -f output_file
```

In this example, if the NLS_LANG environment variable is not set, or is set to *language_territory*.UTF8, then the output LDIF file will contain BASE64-encoded UTF-8 strings for any multibyte characters.

To reload this LDIF file into the directory by using ldapaddmt, use the following syntax:

```
ldapaddmt -h my_host -p port_number -f output_file
```

In the above case, the -E argument is not required because the decoded BASE64 strings are already UTF-8-encoded and can be readily sent to the server.

If the NLS_LANG environment variable is set to a character set other than UTF-8—for example, ".ZHS16GBK"—then the output LDIF file will contain a BASE64 encoded value of simplified Chinese (.ZHS16GBK) strings.

To reload this LDIF file into the directory using ldapaddmt, use the following syntax:

```
ldapaddmt -h host -p port -E ".ZHS16GBK" -f my_input_file.LDIF
```

In the above case the -E argument is required because the decoded BASE64 strings are simplified Chinese, which need to be converted to UTF-8 strings before being sent to the server.

Using Globalization Support with bulkdelete

Add `-encode ".character_set"` to the command.

For example:

```
bulkdelete.sh -connect net_service_name -encode ".ZHS16GBK" -base
"ou=manufacturing,o=acme,c=us"
```

In this case the value for the `-base` option could be in the `ZHS16GBK` native character set, that is, simplified Chinese.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Using Globalization Support with bulkmodify

Add `-E ".character_set"` to the command the argument.

For example:

```
bulkmodify.sh -c my_service_name -E ".ZHS16GBK" -b
"ou=manufacturing,o=acme,c=us" -r title -v Foreman -f "objectclass=*"
```

In this example, values for the `-b`, `-v`, and `-f arguments` can be specified using the simplified Chinese character set.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

9

The Delegated Administration Service

The Delegated Administration Service frees global directory administrators for the more important and complex tasks of directory management. It does this by enabling:

- End users to modify their own passwords without the intervention of an administrator
- Delegated administrators, such as non-technical managers, to create and manage both users and groups
- All users to search parts of the directory to which they have access

This chapter contains these topics:

About the Delegated Administration Service

This section contains these topics:

Delegated Administration Service Units

The Delegated Administration Service is a set of individual, pre-defined services—called Delegated Administration Service units—for performing directory operations on behalf of a user. It makes it easier to develop and deploy administration solutions for both Oracle directory-enabled applications and other directory-enabled applications that use Oracle Internet Directory.

Delegated Administration Service units perform operations such as create user, create group, entry lookup, and change user password. They perform operations on behalf of the application and provide a user interface for displaying the results of those operations.

Delegated Administration Service units are invoked by way of URLs that are published in the directory. To invoke a DAS unit, an application searches for the corresponding URL in the directory.

Users may define their own specialized services to plug into the existing Delegated Administration Service framework.

The Oracle Internet Directory Self-Service Console

Oracle Internet Directory also includes a pre-built, Delegated Administration Service-based web application, called the Oracle Internet Directory Self-Service Console. This application enables administrated access to application data that is managed in the directory. This application enables:

- End users to perform self service on data they are authorized to manage. For example, they can use the Oracle Internet Directory Self-Service Console to change passwords, personal data such as telephone number or office location, or application preferences.

- Subscriber administrators to:
  - Manage subscriber-level information—for example, changing subscriber configurations
  - Provision new users and groups
  - Manage user-level and group-level information within a subscriber—for example, creating user and group entries and editing them

  Subscriber administrators can also use the Self-Service Console to perform such directory operations as:
  - Controlling white pages application access
  - Administering directory attributes not associated with a particular application—for example, telephone number or office location
- Site administrators to:
  - Manage site level information such as site configurations
  - Manage subscriber level information such as creating new subscribers, changing their access privileges and administration (?) privileges.

Benefits of the Delegated Administration Service and the Oracle Internet Directory Self-Service Console

The benefits of using the Delegated Administration Service and the Oracle Internet Directory Self-Service Console include:

- Rapid development and deployment of directory-enabled applications:

  By using Delegated Administration Service units, you can more easily develop the tools your applications need to administer the directory. These units provide most of the functionality applications require.
- Secure access to the directory:

  The Delegated Administration Service uses the **proxy user** feature of Oracle Internet Directory to perform various operations on behalf of users. When you use Delegated Administration Service units to build administration applications, your applications take advantage of this feature. This centralizes the proxy access in one place and improves the directory security. You, as the directory administrator, no longer need to provide super user access to the various directory administration tools that applications require.

- Application user ease-of-use:

  Users of multiple directory-enabled applications interface with a single set of services for administering application-related directory data.

- Ability for sites to delegate directory data administration:

  The Delegated Administration Service allows you to delegate the administration of defined directory data to subscriber administrators and application end users. This makes it easier for sites to manage directory data.

Concepts and Architecture of the Delegated Administration Service

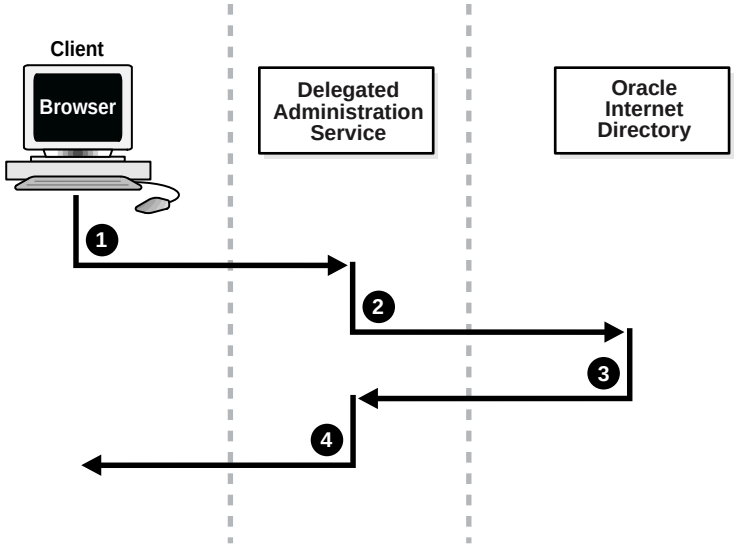
The Delegated Administration Service uses an Oracle HTTP Server that is enabled for small Java programs, called servlets. Together, the Oracle HTTP Server and the servlets

1. Receive requests from clients
2. Process those requests—by either retrieving or updating data in Oracle Internet Directory—and compile the LDAP result into an HTML page
3. Send the HTML page back to the client Web browser

How the Delegated Administration Service Works

Figure 9–1 shows the relationship between components in the Delegated Administration Service environment.

Figure 9–1 Components of the Delegated Administration Service



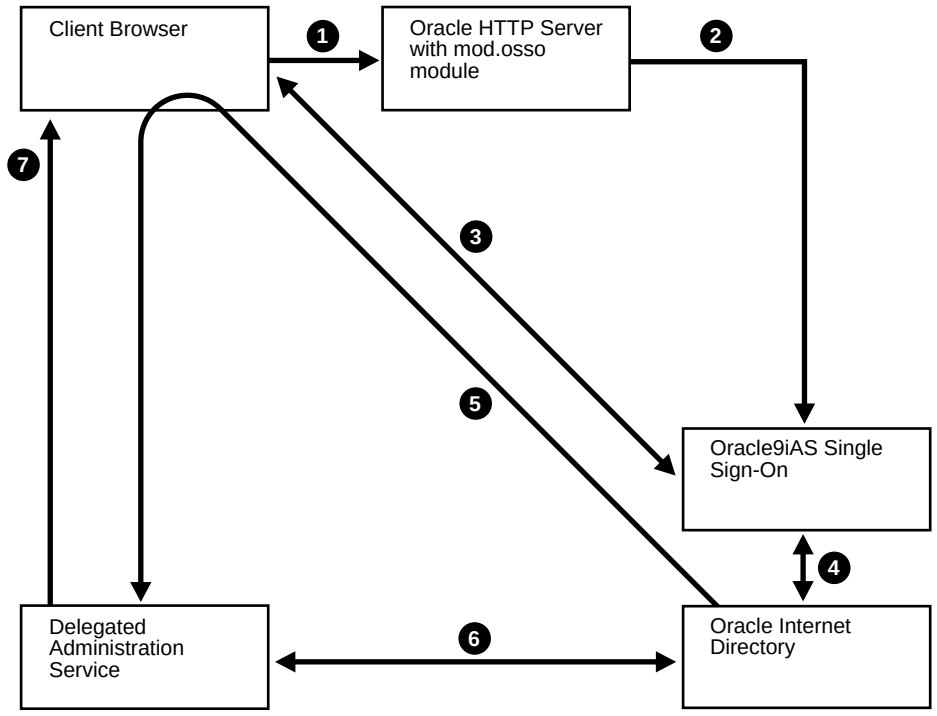
1. The user, from a browser and using HTTP, sends to the Delegated Administration Service a request containing a query to Oracle Internet Directory.
2. The Delegated Administration Service receives the request and launches the appropriate servlet. This servlet interprets the request, and sends it Oracle Internet Directory by using LDAP.
3. Oracle Internet Directory sends the LDAP result to the Delegated Administration Service.
4. The Delegated Administration Service compiles the LDAP result into an HTML page, and sends it to the client Web browser.

The Delegated Administration Service and Oracle9*i*AS Single Sign-On

You can use the Delegated Administration Service in conjunction with Oracle9*i*AS Single Sign-On.

Figure 9–2 shows the relationship between components of the Delegated Administration Service during a search operation within the Oracle9*i*AS Single Sign-On environment.

Figure 9–2 Delegated Administration Service and Oracle9iAS Single Sign-On



1. The user seeks access to the Delegated Administration Service by way of the Oracle HTTP Server with the mod.osso module.
2. If this is the first time during a session that the user is accessing the Delegated Administration Service, then the Oracle HTTP Server transparently directs the user to the Oracle9*i*AS Single Sign-On server for authentication.
3. Oracle9*i*AS Single Sign-On, by way of the Oracle HTTP Server, prompts the user for user name and password. The user provides user name and password.
4. Oracle9*i*AS Single Sign-On verifies the user's credentials by comparing the values the user entered with the corresponding ones stored in Oracle Internet Directory.

5. If it successfully verifies the user name and password, then Oracle9*i*AS Single Sign-On directs the user to the Delegated Administration Service. It also sends to the Delegated Administration Service an encrypted parameter containing the user identifier.
6. The Delegated Administration Service trusts the authentication of the user by Oracle9*i*AS Single Sign-On.

   To enable the user to access the directory, the Delegated Administration Service:

   - Logs in to Oracle Internet Directory on the end user's behalf as a **proxy user**, which has the privilege to switch identities
   - Performs a second bind to the directory, this time using the DN of the end user.

   When the Delegated Administration Service logs in to the directory server by using the DN of the end user, the directory server:

   - Recognizes this second bind as an attempt by the proxy user to switch to the end user's identity
   - Trusts the authentication granted to the end user by the Delegated Administration Service
   - Allows this second bind to succeed without requiring the end user's password.

     See Also: "Authentication" on page 11-4 for information about proxy users and indirect authentication

7. The Delegated Administration Service retrieves the LDAP result from Oracle Internet Directory.
8. The Delegated Administration Service compiles the LDAP result into an HTML page, and sends it to the client Web browser.

Starting and Stopping the Delegated Administration Service

Start the Service by entering:

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

Stop the Service by entering:

```
$ORACLE_HOME/opmn/bin/opmnctl stopall
```

Installing and Configuring the Delegated Administration Service

This section contains these topics:

Log Files for Components in the Delegated Administration Service Environment

Table 9–1 tells you where to find the log files for components in the Delegated Administration Service environment.

Table 9–1 Log Files for Components In Delegated Administration Service Environment

Application	Log File Location
Oracle HTTP Server	$*ORACLE_HOME*/Apache/Apache/logs
Oracle Container for Java (OC4J)	$*ORACLE_HOME*/opmn/logs
Delegated Administration Service	$*ORACLE_HOME*/ldap/log/das.log

To install and configure the Delegated Administration Service, perform the tasks in these sections:

- Task 1: Install the Delegated Administration Service
- Task 2: Verify that the Delegated Administration Service Is Running
- Task 3: Configure the Default Subscriber Context
- Task 4: Configure User Entries

Task 1: Install the Delegated Administration Service

The Delegated Administration Service is installed along with Oracle Internet Directory Release 9.0.2. If you want to enable Oracle9*i*AS Single Sign-On, then you must install and configure the Oracle9*i*AS Single Sign-On Server.

See Also:

- Installation documentation for Oracle Internet Directory Release 9.0.2 for your operating system
- *Oracle9iAS Single Sign-On Administrator's Guide*in the Oracle Database Documentation Library

Task 2: Verify that the Delegated Administration Service Is Running

To verify that the Delegated Administration Service is running, follow these steps:

Step 1: Verify that the Oracle HTTP Server Is Running

To do this, use the following command:

```
ps -ef | grep http
```

See Also:

- Table 9–1 on page 9-8 to find log file locations for components in the Delegated Administration Service environment
- "Starting and Stopping the Delegated Administration Service" on page 9-8

Step 2: Verify that Java (OC4J JVM) Is Running

Use the following command to do this:

```
ps -ef | grep java
```

Be sure that the Java process is running. If it is not, then consult the log file.

> **See Also:** Table 9–1 on page 9-8 for the location of the log file

Step 3: Verify that the Delegated Administration Service Is Running

Using any browser, enter:

`http://`*`host_name`*`:`*`port_number`*`/oiddas/`

where *`host_name`* is the name of the computer on which the Oracle HTTP Server is running. This displays the Delegated Administration Service home page.

Task 3: Configure the Default Subscriber Context

After you have installed the Delegated Administration Service, you may configure the default subscriber context—that is, the root entry of the naming context that contains all entries for the default subscriber.

To configure the default subscriber:

1. Login as the administrator. The default administrative user name is `orcladmin`, and the default password is `welcome`.
2. Select the Configuration tab.
3. In the Directory Configuration section:
   a. In the Attribute for Login Name field, enter the attribute by which you want users to identify themselves when they log in—for example: cn, UID, EmployeeNumber, SSN.
   b. In the User Search Base Context field, enter the DN of the entry under which the user entries for this subscriber are located.
   c. In the Group Search Base Context field, enter the DN of the entry under which group entries for this subscriber are located.
   d. In the Search Return Limit field, enter the number of entries you want displayed in the search results.

4. In the Logo Management section:
   a. If you want to display the subscriber's logo in the upper left corner of the Delegated Administration Service user interface, then select the Enable Subscriber Logo checkbox. Otherwise, leave it unselected.
   b. If you want to display the product name, namely Internet Directory, in the upper left corner of the Delegated Administration Service user interface, then select the Enable Product Logo checkbox. Otherwise, leave it unselected.
   c. In the Update Subscriber Logo field, enter the path and file name of this subscriber's logo, or, alternatively, navigate to it by choosing Browse.
5. When you have entered the location of the corporate image logo file, choose Submit to save your changes.

Task 4: Configure User Entries

When a user creates or edits a user entry, the user interface displays various categories—including, for example, basic information, password, and photo—each with its own set of attributes. You can customize the way the Delegated Administration Service displays these categories and the corresponding attributes.

Specifically, the Delegated Administration Service enables you to:

- Add object classes to user entries, and add and modify their attributes
- Specify the categories of attributes you want to enable users to add or modify
- Customize the way the Delegated Administration Service displays those categories and attributes

To configure user entries:

1. Select the Configuration tab, then choose User Entry. This displays the Configure User Object Classes window listing the existing object classes for user entries.
2. To add an object class for user entries:
   a. Choose Add Object Class. This displays the All Object Classes window.
   b. Select an object class you want to add, then choose Add. This returns you to the Specify Object Class window. The object class you just chose is now listed as an existing object class.
   c. To add more object classes, repeat these steps.

If you are satisfied with the object classes, then choose Next to display the Configure Attributes window.

3. To add attributes or modify the way the Delegated Administration Service displays those attributes:

   a. Choose Add New Attribute to display the Add New Attribute window.

   b. From the Directory Attribute Name combo box, select the attribute you want to add.

   c. Enter values for the fields as described in Table 9–2.

Table 9–2 Fields in the Configure Attributes Window

Field	Description
UI Label	Specify the friendly name of the attribute in the user interface. For example, you can display the sn attribute as *Last Name* in the interface.
Required	Specify whether you want the attribute to be required. Required attributes appear in the interface with an asterisk (*) to the left of the field. If you do not select this check box, then the attribute is optional.
Viewable	Specify whether you want the attribute to appear in search results by selecting this check box.
UI Type	Specify the type of interface for this field. Options are: ■ Single Line Text—a text field into which the user enters a value ■ Multi Line Text—a text area where a user can type multiple lines of text ■ Predefined List—a combo box in which a user selects a value from a drop-down list. To specify values for the drop-down list, choose Edit to display the Editing Attribute window. In the LOV Values text area, enter each value, then press the ENTER key. ■ Date—a text field into which the user enters a date—for example, an employee's birthday ■ Browse and Select—a button enabling the user to browse for a manager's entry or any entry that needs a DN as an attribute value ■ Number—a text field into which the user enters numbers only—for example, a postal code

d. Choose Done to return to the Configure User Attributes window. The attribute you just chose is now listed in the attribute list.

If you are satisfied with the user attributes, then choose Next to display the Create Attribute Categories window.

4. Use the Create Attribute Categories window to customize the way that categories of attributes are displayed to a user.

To add a new category:

a. Choose Add New Category.

b. In the UI Label field, enter the friendly name of the category—for example, Telephone Numbers or Organizational Details.

c. Choose Done to return to the Create Attribute Categories window.

To modify a category:

a. In the Select column, select the appropriate category.

b. In the UI Label and Display Order columns, edit the appropriate fields. To designate the display order, specify the category you want to appear at the top of the window with a 0, the next with a 1, the next with a 2, and so on.

To delete a category, select it, then choose Delete.

If you are satisfied with the attribute categories, choose Next to display the Configure Attribute Categories window.

5. To configure each category of attributes, use the Configure Attribute Categories window. For each category, it displays two lists:

- All Attributes—All attributes available for this category
- Selected Attributes—The attributes in this category that you want to enable users to modify.

To configure each attribute category:

a. Move items between the two lists by selecting one or more at a time, then choosing the appropriate arrow.

b. Within the Selected Attributes list for each category, set the attribute display order by using the up and down arrow buttons on the right of the list.

When you have finished configuring attribute categories, choose Next to display the Configure Public Groups window.

6. To configure the display of public group lists in the Delegated Administration Service user interface:

   To enable users to assign users to public groups, select the Enable Public Group assignment check box. Otherwise, leave it unselected.

   To add a public group, choose the Add Group button to display the Search and Select: Public Groups window. In the Group Name Begins With field, enter the first few letters of the name of the group you want to add, select it in the table of search results, then choose Select.

   To delete a public group, select the group from the table and choose Delete.

Searching for User and Group Entries by Using the Delegated Administration Service

This section contains these topics:

- Searching for User Entries by Using the Delegated Administration Service
- Searching for Group Entries by Using the Delegated Administration Service

Searching for User Entries by Using the Delegated Administration Service

To search for users:

1. Select the Directory tab, then select Users.
2. In the Search for User field, enter the first few characters of the name of the user. For example, if you are searching for Anne Smith, you could enter Ann.
3. Choose Go to display the search results.

Searching for Group Entries by Using the Delegated Administration Service

To search for groups:

1. Select the Directory tab, then select Groups.
2. In the In Search Group Name text box, enter the first few characters of the name of the group for which you are searching.
3. Choose Go to display the entries that match the criteria you entered.

Managing Users, Groups, and Subscribers by Using the Delegated Administration Service

This section contains these topics:

Creating User Entries by Using the Delegated Administration Service

To create a user entry:

1. Select the Directory tab, then select Users.
2. Choose Create to display the Create User window.
3. Enter values in the required and other appropriate fields.
4. Verify that you have entered all information correctly, then choose Submit.

Modifying User Entries by Using the Delegated Administration Service

To modify a user entry:

1. Select the Directory tab, and perform a search for the user whose entry you want to modify.
2. Select the user whose entry you want to modify, then choose Edit to display the Edit User window.
3. Modify values in the required and other appropriate fields, then choose Finish.

See Also: Searching for User Entries by Using the Delegated Administration Service

Deleting User Entries by Using the Delegated Administration Service

To delete a user entry:

1. Select the Directory tab, and perform a search for the user whose entry you want to delete.
2. Select the user whose entry you want to delete, then choose Delete.

Assigning Privileges to Users by Using the Delegated Administration Service

You can privilege a user to do one or all of the following:

- Create and edit users and groups
- Assign privileges to other users and to groups

You can also revoke privileges from a user.

To assign privileges to a user:

1. Select the Directory tab, and perform a search for the user entry to which you want to assign privileges.
2. Select the user to whom you want to assign privileges, then choose Assign Privilege to display a list of privileges.
3. Select the privileges you want to assign to this user. Options are:

Privilege	Description of Access Granted
Allow user creation	Create user entries
Allow user editing	Modify user entries
Allow user deletion	Delete user entries
Allow group creation	Create group entries
Allow group editing	Modify group entries
Allow group deletion	Delete group entries
Allow privilege assignment to users	Assign access rights to users

Privilege	Description of Access Granted
Allow privilege assignment to groups	Assign access rights to groups
Allow Delegated Administration Service configuration	Configure Delegated Administration Service user interface
Allow resource management for Oracle Reports and Forms based applications	Configure resource types and set up default resource access information for Oracle Reports and Forms based applications

4. Choose Submit, or, to assign privileges to another user, choose Specify Other User and repeat the process.

Creating Group Entries by Using the Delegated Administration Service

To create group entries:

1. Select the Directory tab, select Groups, then select Create. This displays the Create Group window.
2. In the Basic Information section, in the Name field, enter the name for this group.
3. In the Display Name field, enter the friendly name. For example, if the **RDN** is `OracleDBCreators`, then you could enter the display name as Oracle Database Creators.
4. In the Description field, enter a brief description of this group.
5. To hide this group entry from all but its owners, in the Group Visibility field, select Hidden. Otherwise, accept the default, Not Hidden.

   Choose Next. This displays the User Members page.
6. The creator of the group is automatically a group owner. To specify an additional owner of this group:
   a. In the Owners section, choose Add Owner to display the Search and Select: User window.
   b. Perform a search for the entry of the user you want to specify as an owner of the group, then choose Select. This returns you to the Create Group window. The user you specified is listed in the Owners section.

To remove an owner, in the Owners section, select the owner's name and choose Remove.

7. To add a user as a member of this group:
   a. In the Members section, choose Add User Member to display the Search and Select window.
   b. Perform a search for the entry of the user you want to specify as a member of this group, then choose Select. This returns you to the Create Group window. The user you specified is listed in the User Members section.

   To remove a user from this group, in the Add User Members section, select the user's name and choose Remove.
8. To add a group as a member of this group:
   a. In the Members section, choose Add Group Member to display the Search and Select window.
   b. Perform a search for the entry of the group you want to specify as a member of this group, then choose Select. This returns you to the Create Group window. The group you specified is listed in the Members section.

Modifying Group Entries by Using the Delegated Administration Service

To modify group entries:

1. Select the Directory tab and perform a search for the group entry you want to modify.
2. Select the group entry you want to modify, then choose Edit to display the Edit Group window.
3. Modify the fields as described in "Creating Group Entries by Using the Delegated Administration Service" on page 9-17, then choose Finish.

Deleting Group Entries by Using the Delegated Administration Service

To delete group entries:

1. Select the Directory tab, and perform a search for the group whose entry you want to delete.
2. Select the group whose entry you want to delete, then choose Delete.

Assigning Privileges to Groups by Using the Delegated Administration Service

You can privilege a group to do one or more of the following:

- Create and edit new users and groups
- Assign privileges to users and to other groups

To assign privileges to groups:

1. Select the Directory tab, choose Groups, and perform a search for the group entry to which you want to assign privileges.
2. Select the group to which you want to assign privileges, then choose Assign Privilege to display a list of privileges.
3. Select the privileges you want to assign to this group. Options are:

Privilege	Description of Access Granted
Allow user creation	Create user entries
Allow user editing	Modify user entries
Allow user deletion	Delete user entries
Allow group creation	Create group entries
Allow group editing	Modify group entries
Allow group deletion	Delete group entries
Allow privilege assignment to users	Assign access rights to users
Allow privilege assignment to groups	Assign access rights to groups
Allow Delegated Administration Service configuration	Configure Delegated Administration Service interface
Allow resource management for Oracle Reports and Forms based applications	Configure resource types and set up default resource access information for Oracle Reports and Forms based applications

4. Choose Submit, or, to assign privileges to another user, choose Specify Other Group and repeat the process.

Changing Passwords by Using the Delegated Administration Service

You can change your own password and, if you have the privilege to modify user or group entries, then you can change another user's or a group's password.

Changing Your Own Password

You can change the password you use for authenticating to Oracle9*i*AS Single Sign-On, the Delegated Administration Service, the Enterprise Security Manager, and Oracle Portal. You can also change your password for other Oracle components.

To change your password:

1. Login to the Delegated Administration Service and select the My Profile tab.
2. Select Change My Password.

   To change your password to Oracle9*i*AS Single Sign-On, the Delegated Administration Service, the Enterprise Security Manager, and Oracle Portal:

   a. In the Single Sign-On section, in the Old Password field, enter your current password.

   b. In the New Password field, enter your new password, then confirm it in the Confirm New Password field.

   c. Choose Submit.

   To change your password to another Oracle component:

   a. In the Application Passwords section, select the Oracle component for which you want to specify a new password.

   b. Choose Update Password to display the Change Application Password window.

   c. In the New Password field, enter your new password, then confirm it in the Confirm New Password field.

   d. Choose Submit.

Changing Another User's Password

You can change another user's password if you have the necessary access rights. To change another user's password:

1. Select the Directory tab, and perform a search for the entry of the user whose password you want to change.

2. Select the user entry, then choose Edit to display the Edit User window.
3. In the Basic Information section, enter, then confirm, the password you want to assign to the user.
4. Choose Submit.

10

Attribute Uniqueness

This chapter explains attribute uniqueness in Oracle Internet Directory.

This chapter contains these topics:

Introduction

In the prior Oracle Internet Directory architecture, the only way to enforce attribute uniqueness was to make an attribute a part of your DN. This worked well with the user identifier (if used as the RDN), but it was not always appropriate and easy to configure. Within a level of a branch of the tree, it was guaranteed to be unique. For example, if your DN was `uid=dlin, ou=people, o=oracle`, then this would be unique directly under `ou=people`. However, you could have the same user identifier in another branch—for example, `uid=dlin, ou=others, o=oracle`. In short, attribute uniqueness was guaranteed only under a given branch, and only within one level.

The applications Oracle Internet Directory synchronizes with can use attributes other than DN as their unique keys. The ability of Oracle Internet Directory to enforce attribute uniqueness enables all applications their own notions of "user," to synchronize their user base with a user repository stored in an enterprise's Oracle Internet Directory server.Attribute uniqueness implements the checks that ensure the values of specified attributes are unique each time an entry is modified.

Attribute uniqueness enables the user to define attribute uniqueness across the following:

- The entire directory
- One subtree
- One object class

Concepts

The attribute uniqueness constraint is similar to the pre-operation trigger. This means that the directory server checks all update operations before it performs an LDAP operation. The directory server determines whether the operation applies to an attribute and a suffix (subtree) that you have configured the directory server to monitor.

If an update operation applies to an attribute and suffix monitored by the directory server, and the update operation would cause two entries to have the same attribute value, then the server terminates the operation and returns a constraint violation error to the client.

The directory server performs attribute uniqueness checks on the following:

- A single attribute
- One subtree per attribute
- One object class

> **Note:** Attribute uniqueness only works on cataloged attributes.

To check uniqueness of several attributes, you must create a separate instance of the uniqueness constraint for each attribute you want to check.

There are different ways of configuring attribute uniqueness constraint as follows:

- Enable the user to define attribute uniqueness across the entire directory
- Enable the user to define attribute uniqueness across one subtree per attribute

  For example, Oracle hosts the directories for Company1 and Company2. When you add an entry such as `uid=dlin,ou=people,o=Company1,dc=oracle,dc=com`, you must enforce uniqueness only in the `o=Company1,dc=oracle,dc=com` subtree. Do this by listing the DN of the subtree explicitly in the attribute uniqueness constraint configuration.

- Enable the user to define attribute uniqueness across one object class

  For example, ID is a unique attribute on the object class machine, and it is also a unique attribute on the object class person. With the attribute uniqueness enforcement in Oracle Internet Directory, attempting to load two machines with same IDs or two people with same IDs into Oracle Internet Directory returns an attribute uniqueness constraint violation error to the client. A machine ID can have the same value as a person's ID.

Requirements

This section explains requirements for attribute uniqueness

This section contains these topics:

Creating Attribute Uniqueness

To ensure that a particular attribute in your directory always has unique values, you must create an instance of attribute uniqueness for the attribute you want to check. For example, to ensure that every entry in your directory that includes a mail attribute has a unique value for that attribute, you must create an instance of attribute uniqueness associated with mail.

For the same attribute, if there are two different uniqueness policies associated with the attribute and the scope of one policy is a subset of the other scope, then the outermost (or higher level) of the policies take precedence.

Creating Attribute Uniqueness Across an Entire Directory

To create an instance of attribute uniqueness across an entire directory, the required input information is an attribute name for which you want to enforce value uniqueness.

Creating Attribute Uniqueness Across One Subtree

To create an instance of attribute uniqueness across one or more subtrees, the required input information is as follows:

- Attribute name for which you want to enforce value uniqueness
- Subtree locations under which the uniqueness constraint is to be enforced

Creating Attribute Uniqueness Across One Object Class

To create an instance of attribute uniqueness across one object class, the required input information is as follows:

- Attribute name for which you want to enforce value uniqueness
- Object class name

Enabling and Disabling Attribute Uniqueness

You can enable or disable attribute uniqueness.

Enabling Attribute Uniqueness

Use the `ldapmodify` command-line tool to change the state of the attribute uniqueness policy to be on. If you modify the attribute uniqueness constraint, then you must restart the directory server.

Disabling Attribute Uniqueness

Use the `ldapmodify` command-line tool to change the state of the attribute uniqueness policy to be off. If you remove the attribute uniqueness constraint, then you must restart the directory server.

Specifying the Subtree

Users can specify the suffix or subtree under which they want the checking to ensure attribute uniqueness by modifying the subtree location attribute in the policy object.

The user can use the `ldapmodify` command-line tool to import the LDIF file that contains update statements into the directory.

Note: Users must restart the directory server to enable the modified policy.

Deleting an Attribute Uniqueness Policy

Use the `ldapdelete` command-line tool to delete an attribute uniqueness policy.

Note: The directory server must be restarted to disable the policy after removing the policy.

Configuration Interface

As Table 10–2, "Attribute Uniqueness Constraint Entry" illustrates, each attribute uniqueness constraint entry has the following attributes.

Table 10–1 Attribute Uniqueness Constraint Entry Attributes

Attribute	Description
`orcluniqueattrname`	The user must specify this attribute.

`orcluniquescope`	The user has the option to specify this attribute with one of the following values:
	`base`
	`onelevel`
	`sub`
	If this attribute is not specified, `sub` is used by default.

`orcluniquesubtree`	The user can specify the subtree where the attribute uniqueness constraint is enforced. By default, it enforced from the root directory.
`orcluniqueobjectclass`	The user can specify the object class where the attribute uniqueness constraint is enforced. By default, it is enforced on all object classes.

Defined Policy Location and Model

All attribute uniqueness constraint entries must be stored under `cn=unique, cn=Common, cn=Products, cn=OracleContext`.

As Table 10–2 illustrates, the default value is applied respectively, when orclunqiuescope, orcluniquesubtree, or orcluniqueobjectclass is not specified in the attribute uniqueness constraint entry. By default, orcluniquescope is subtree, orcluniquesubtree is the entire directory, and orcluniqueobjectclass is all object classes.

Policy Scoping Rules

When multiple attribute uniqueness constraints have different values in orcluniqueattrname, their effects are independent of each other.

When multiple attribute uniqueness constraints have the same values in orcluniqueattrname, but different values in orcluniquesubtree, and their subtrees overlap, the attribute uniqueness constrain with largest subtree scope is in effect.

When multiple attribute uniqueness constraints have the same values in orcluniqueattrname and orcluniquesubtree, but different values in orcluniquescope, the attribute uniqueness constraint with the largest search scope is in effect.

When multiple attribute uniqueness constraints have the same values in orcluniqueattrname, orcluniquesubtree, and orcluniquescope, but different values in orcluniqueobjectclass, then the union of attributes belonging to those object classes is checked.

When multiple attribute uniqueness constraints have the same values in orcluniqueattrname and orcluniqueobjectclass but different values in orcluniquesubtree, and their subtrees overlap, the attribute uniqueness constraint with largest subtree scope is in effect.

When multiple attribute uniqueness constraints have the same values, respectively, in orcluniqueattrname, orcluniquesubtree and, orcluniqueobjectclass, the attribute uniqueness constraint with largest scope is in effect.

Table 10–2 Attribute Uniqueness Constraint Entry

Attribute Name	Must Specify	Valid Value	Default Value	Default Effect
orcluniqueattrname	Yes	Any string	N/A	N/A
orcluniquescope	No	One of the following:	sub	

Table 10–2 Attribute Uniqueness Constraint Entry

Attribute Name	Must Specify	Valid Value	Default Value	Default Effect
		base		Search base only
		onelevel		Search one level
		sub		Search subtrees
orcluniquesubtree	No	Any string	""	Entire directory
orcluniqueobjectclass	No	Any string	""	All object classes

Applying the Attribute Uniqueness Feature

The following example applies the attribute uniqueness feature through Oracle Internet Directory.

Scenario: We want to make sure all employee IDs are unique for all US employees at Oracle Corporation.

Solution: Follow these steps to create and apply an attribute uniqueness constraint.

1. Create an attribute uniqueness constraint entry (in LDIF format) as follows:

```
dn: cn=constraint1, cn=unique, cn=common, cn=products, cn=oraclecontext
objectclass: orclUniqueConfig
orcluniqueattrname: employeenumber
orcluniquesubtree: o=Oracle Corporation, c=US
orcleuniqueobjectclass: person
```

2. To apply the attribute uniqueness feature, the attribute uniqueness constraint entries must be loaded using the following:

```
ldapadd –h <host> -p <port> -D <dn> -w <password> -f constraint1.dat
```

3. Restart the directory server.

Now, LDAP enforces attribute uniqueness on the `employee ID` of all US employees at Oracle Corporation.

To remove this constraint do the following:

1. Delete the attribute uniqueness entry.

2. Restart the directory server.

Known Limitations

Regarding replication and attribute uniqueness, when an attribute uniqueness constraint is present in the Oracle Internet Directory replication environment, be very careful about configuring the attribute uniqueness constraints on each server.

Simple Replication Scenario

All modifications by client applications are performed on the supplier server, therefore, the attribute uniqueness constraint should be enabled on the supplier server. It is not necessary to enable the attribute uniqueness constraint on the consumer server.

Enabling the attribute uniqueness constraint on the consumer server does not prevent the Oracle Internet Directory server from operating correctly, but is likely to cause a performance degradation.

Multimaster Replication Scenario

In a multi-master replication scenario, the two masters act both as suppliers and consumers of the same replica. Multi-master replication uses a loosely consistent replication model. Enabling an attribute uniqueness constraint on one of the servers does not ensure that attribute values are unique across both masters at any given time. Enabling an attribute uniqueness constraint on only one server can cause inconsistencies in the data held on each replica.

The attribute uniqueness constraint must be enabled on both masters.

However, there may still be an inconsistent state. For example, in both masters we can successfully modify entries to the same attribute value. Later, when the changes are replicated to the other node, the conflict becomes apparent. You must take this type of conflict resolution into consideration as well. That is, there is a decision to be made about conflict resolution. Should conflict resolution be the replication server's responsibility or not?

Part III

Directory Security

This part contains discusses the features that enable you to secure data within the directory, as well as how to establish access controls for administering applications in enterprises and hosted environments. It contains these chapters:

11

Directory Security Concepts

This chapter describes the security features available with Oracle Internet Directory. It contains these topics:

Data Integrity

Oracle Internet Directory ensures that data has not been modified, deleted, or replayed during transmission by using Secure Sockets Layer (SSL). This SSL feature generates a cryptographically secure message digest—through cryptographic checksums using either the **MD5** algorithm or the **Secure Hash Algorithm (SHA)**—and includes it with each packet sent across the network.

> **See Also:** Chapter 12, "Secure Sockets Layer (SSL) and the Directory" for more information about SSL

Data Privacy

Oracle Internet Directory ensures that data is not disclosed during transmission by using **public-key encryption** available with SSL. In public-key encryption, the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the recipient decrypts the message using the recipient's private key. Specifically, Oracle Internet Directory supports two levels of encryption available through SSL:

- DES40

  The DES40 algorithm, available internationally, is a variant of **DES** in which the secret key is preprocessed to provide forty effective **key** bits. It is designed for use by customers outside the USA and Canada who want to use a DES-based encryption algorithm. This feature gives commercial customers a choice in the algorithm they use, regardless of their geographic location.

- RC4_40

  Oracle has obtained license to export the RC4 data encryption algorithm with a 40-bit key size to virtually all destinations where other Oracle products are available. This makes it possible for international corporations to safeguard their entire operations with fast cryptography.

> **See Also:** Chapter 12, "Secure Sockets Layer (SSL) and the Directory" for more information about SSL

Authorization

Authorization is the process of ensuring that a user reads or updates only the information for which that user has privileges. When directory operations are attempted within a directory session, the directory server ensures that the user has

the requisite permissions to perform those operations. If the user does not have the requisite permissions, then the directory server disallows the operation. Through this mechanism, the directory server protects directory data from unauthorized operations by directory users. This mechanism is called access control.

Access control information is the directory metadata that captures the administrative policies relating to access control. This information is stored in Oracle Internet Directory as user-modifiable operational attributes, each of which is called an **access control item (ACI)**.

Typically, a list of these ACI attribute values, called an **access control list (ACL)**, is associated with directory objects. The attribute values on that list govern the access policies for those directory objects.

Access control information associated with a directory object represents the permissions on the given object that various directory user entities (or subjects) have. Thus, an ACI consists of:

- The object to which you are granting access
- The entities or subjects to whom you are granting access
- The kind of access you are granting

Access control policies can be prescriptive, that is, their security directives can be set to apply downward to all entries at lower positions in the **directory information tree (DIT)**. The point from which such an access control policy applies is called an **access control policy point** (**ACP**).

ACIs are represented and stored as text strings in the directory. These strings must conform to a well defined format, called the ACI directive format. Each valid value of an ACI attribute represents a distinct access control policy.

The following features of directory access control can be used by applications running in a hosted environment.

- Prescriptive access control

  Enables the service provider to specify access control lists (ACLs) for a collection of directory objects, instead of having to state the policies for each individual object. This feature simplifies the administration of access control, especially in large directories where many objects are governed by identical or similar policies.

- Hierarchical access control administration model

  Enables the service provider to delegate directory administration to subscribers. The subscriber could in turn delegate further if necessary.

- Administrative override control for delegated domains

  Enables the service provider to perform diagnosis and recovery from unintentional account lockout or accidental security exposure.

- Dynamic evaluation of access control entities

  Enables subtree administrators to identify both subjects and objects in terms of their namespace and their association with other objects in the directory. For example, the administrator of one subscriber subtree can allow only a user's manager to update that user's salary attribute. The administrator of another subscriber subtree can establish and enforce a different policy regarding salary attributes.

Authentication

Authentication is the process by which the directory server establishes the true identity of the user connecting to the directory. It occurs when an LDAP session is established by means of the ldapbind operation. Thus every session has an associated user identity.

To verify the identities of users, hosts, and clients, Oracle Internet Directory enables two general kinds of authentication: direct and indirect.

Direct Authentication

There are three direct authentication options:

- Anonymous Authentication

  When users authenticate anonymously, they simply leave the user name and password fields blank when they log in. Each anonymous user then exercises whatever privileges are specified for anonymous users.

- Simple Authentication

  When using simple authentication, the client identifies itself to the server by means of a DN and a password that are not encrypted when sent over the network.

- Secure Sockets Layer (SSL) Authentication

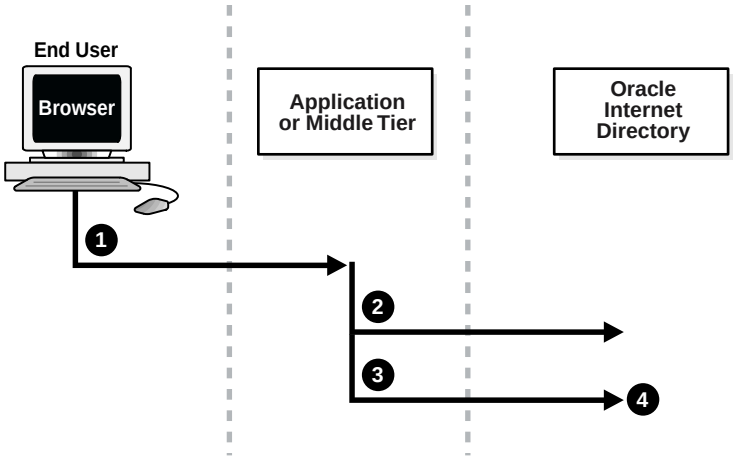
  This involves the exchange of certificate**s** issued by trusted certificate authorities.

Indirect Authentication

Indirect authentication occurs through any entity that has credentials in the directory—for example, an application such as the Delegated Administration Service, or a middle tier such as a firewall or a RADIUS server. The application or middle tier becomes a **proxy user** that impersonates an end user, performing directory operations on that end user's behalf.

Figure 11–1 and the accompanying text explain how indirect authentication takes place.

Figure 11–1 Indirect Authentication



Indirect authentication takes place as follows:

1. The end user sends to the application or middle tier a request containing a query to Oracle Internet Directory. The application or middle tier authenticates the end user.
2. The application or middle tier binds to the directory.
3. The application or middle tier performs a second bind, this time using the DN of the end user. It does not enter the end user's password.
4. The directory server recognizes this second bind as an attempt by the application or middle tier to switch to the end user's identity. It trusts the authentication granted to the end user by the application or middle tier, but must verify that the application or middle tier has the right to be the proxy for

this user. It checks to see whether the ACP governing the end user entry gives this application or middle tier the proxy right for this end user.

* If the application or middle tier does have the necessary proxy right, then the directory server changes the authorization identity to that of the end user. All subsequent operations occur as if that end user had connected directly to the server and had been directly authenticated.
* If the application or middle tier does not have the necessary proxy right, then the directory server returns an error message, "Insufficient Access."

See Also: Operations: What Access Are You Granting? on page 13-10

The directory server can, in the same session, authenticate and authorize other end users. It can also switch the session from that of the end user to that of the application or middle tier that opened the session.

To close the session, the application or middle tier sends an unbind request to the directory server.

For example, suppose you have:

- A middle tier that binds to the directory as `cn=User1`, which has proxy access on the entire directory
- An end user that can bind to the directory as cn=User2

When this end user sends to the application or middle tier a request containing a query to Oracle Internet Directory, the application or middle tier authenticates the end user. The middle tier service then binds to the directory by using its own identity `cn=User1`, then performs a second bind, this time by using only the DN of the end user, `cn=User2`. The Oracle directory server recognizes this second bind as an attempt by the proxy user to impersonate the end user. After the Oracle directory server verifies that `cn=user1` has proxy access, it then allows this second bind to succeed. It does not require any further validation of the end-user DN, such as a password. For the rest of the session, all LDAP operations are access-controlled as if `cn=User2` were performing them.

If another user of such an application requests its services while a prior user is being serviced, the application can establish a new connection and proceed as above without disrupting that session. If, however, no prior user is still being serviced, the existing established connection can be re-used again and again without any need for a new connection.

Protection of User Passwords for Directory Authentication

Oracle Internet Directory can protect a user's directory password by storing it in the `userPassword` attribute as a one-way hashed value. You select the hashing algorithm you want to use. Storing passwords as one-way hashed values—rather than as encrypted values—more fully secures them because a malicious user can neither read nor decrypt them.

> **See Also:** "Storing Password Verifiers for Authenticating to Oracle Internet Directory"

Password Policies

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

When you establish a password policy, you set the following types of rules, to mention just a few:

- The maximum length of time a given password is valid
- The minimum number of characters a password must contain
- The number of numeric characters required in a password

> **See Also:** Chapter 18, "Password Policies" for a fuller description of the rules you set when establishing password policies

12

Secure Sockets Layer (SSL) and the Directory

This chapter explains how to configure Secure Sockets Layer (SSL) for use with Oracle Internet Directory. If you use Secure Sockets Layer (SSL), you may also configure strong authentication, data integrity, and data privacy.

This chapter contains these topics:

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

See Also: "Security" on page 2-13 for a conceptual overview of SSL in relation to Oracle Internet Directory

Supported Cipher Suites

A cipher suite is a set of authentication, encryption, and data integrity algorithms used for exchanging messages between network nodes. During an SSL handshake, the two nodes negotiate to see which cipher suite they will use when transmitting messages back and forth.

The Oracle Internet Directory supports the following SSL cipher suites:

Table 12–1 SSL Cipher Suites Supported in Oracle Internet Directory

Cipher Suite	Authentication	Encryption	Data Integrity
SSL_RSA_WITH_3DES_EDE_CBC_SHA	RSA	DES40	SHA
SSL_RSA_WITH_RC4_128_SHA	RSA	RC4_40	SHA
SSL_RSA_WITH_RC4_128_MD5	RSA	None	MD5
SSL_RSA_WITH_DES_CBC_SHA	RSA	None	SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA		3DES_EDE_CBC	SHA
SSL_DH_anon_WITH_RC4_128_MD5		RC4_40	MD5
SSL_DH_anon_WITH_DES_CBC_SHA		DES_CBC	SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5		RC4_40	MD5
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA		DES40	SHA
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5		RC4_40	MD5
SSL_DH_anon_EXPORT_WITH_DES40_CBC_ SHA		DES40	SHA

SSL Client Scenarios

Oracle Internet Directory clients can use SSL 2.0 or SSL 3.0. A client over SSL can connect to a server anonymously or by using either simple or strong authentication.

When both a client and server authenticate themselves to each other, SSL derives the identity information it requires from the X509v3 digital certificates.

Configuring SSL Parameters

During start-up of a **directory 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, set the SSL Enable parameter in the configuration settings to 1: the default secure port is 636. To allow the same instance to run non-secure connections concurrently, set SSL Enable to 2: the default non-secure port is 839.

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. The default set may be required by Oracle Support Services in the diagnosis of certain technical issues.

See Also:

- "Managing Server Configuration Set Entries" on page 5-2 for instructions on how to set these parameters
- "Configuration Set Entry Attributes" on page C-5 for a description of these parameters

Configuring SSL Parameters by Using Oracle Directory Manager

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

Note: You cannot directly change the parameters for an active instance. If you want to change the parameters for an active instance, change the parameters in a configuration set entry and save it. After it is saved, you can stop current instances and refer to the newly modified configuration set in the start server message.

To view and modify SSL configuration parameters:

1. In Oracle Directory Manager's navigator pane, expand Oracle Internet Directory Servers > *directory server* > Server Management.
2. Expand either Directory Server or Replication Server, as appropriate. The numbered configuration sets are listed beneath your selection.
3. Select the configuration set that you want to examine. The group of tab pages for that configuration set entry appear in the right pane.
4. Select the SSL Settings tab page.

   You can change the parameters in this tab page and save them. The fields in this tab page are described in the following table:

Field	Description
SSL / non-SSL Enable	Set 0 for only non-secure operation; default port is 839, changeable below. Set 1 for only SSL authentication; default port is 636, changeable below. Set 2 for both non-secure operation and SSL authentication.
SSL Authentication	Choose one of the following: ■ No SSL Authentication—Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, SSL encryption / decryption only is used. ■ SSL Client and Server Authentication—Both client and server authenticate themselves to each other and send certificates to each other. ■ SSL Server Authentication—Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.

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

SSL Wallet Password	Type the password for the server-side wallet. This password was set during creation of the wallet. If you change the password, you must change this parameter.

Field	Description
SSL Port	The default SSL port is 636. You can change the SSL port.
Non-SSL Port	The default non-SSL port is 839. You can change the non-SSL port.

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

Configuring SSL Parameters by Using Command-Line Tools

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

Issues Specific to This Release of Oracle Internet Directory

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

In Oracle Internet Directory Release 9.0.2, the Oracle directory replication server cannot communicate directly with SSL-enabled Oracle directory server instances.

See Also: Chapter 5, "Oracle Directory Server Administration" for instructions on how to configure server instances

13

Directory Access Control

This chapter provides an overview of access control policies and describes how to administer directory access control by using either Oracle Directory Manager or the command-line tool, ldapmodify.

This chapter contains these topics:

- Overview of Access Control Policy Administration
- Managing Access Control by Using Oracle Directory Manager
- Managing Access Control by Using Command-Line Tools
- How ACL Evaluation Works

See Also:

- "Globalization Support" on page 2-14 for a conceptual explanation before you begin implementing and administering access control policies
- Appendix B, "The Access Control Directive Format" for information about the format or syntax of Access Control Items (ACIs)

Overview of Access Control Policy Administration

You manage access control policies by configuring the values of the **ACI** attributes within appropriate entries. You can do this by using either Oracle Directory Manager or ldapmodify.

This section contains these topics:

- Access Control Management Constructs
- Access Control Information Components

Access Control Management Constructs

This section discusses the structures used for access control in Oracle Internet Directory. These include:

- Access Control Policy Points (ACPs)
- The orclACI attribute for prescriptive access control
- The orclEntryLevelACI attribute for entry-level access control
- Privilege Groups

Access Control Policy Points (ACPs)

ACPs are entries in which the `orclACI` attribute has been given a value. The `orclACI` attribute value represents the access policies that are inherited by the subtree of entries starting with the ACP as the root of the subtree.

When a hierarchy of multiple ACPs exists in a directory subtree, a subordinate entry in that subtree inherits the access policies from all of the superior ACPs. The resulting policy is an aggregation of the policies within the ACP hierarchy above the entry.

For example, if an ACP is established in the HR department entry, and the Benefits, Payroll, and Insurance groups are entries within the HR department, then any entry within those groups inherits the access rights specified in the HR department entry.

When there are conflicting policies within a hierarchy of ACPs, the directory applies well-defined precedence rules in evaluating the aggregate policy.

See Also: "How ACL Evaluation Works" on page 13-47

The orclACI Attribute for Prescriptive Access Control

The `orclACI` attribute contains **access control list (ACL)** directives that are prescriptive—that is, these directives apply to all entries in the subtree below the ACP where this attribute is defined. Any entry in the directory can contain values for this attribute. Access to this attribute itself is controlled in the same way as access to any other attribute.

Note: It is possible to represent ACL directives specific to a single entry in the `orclACI` attribute. However, in such scenarios, for administrative convenience and performance advantages, Oracle Corporation recommends using `orclEntryLevelACI`—discussed in "The orclEntryLevelACI Attribute for Entry-Level Access Control" on page 13-3. This is because the LDAP operational overhead increases with the number of directives represented through `orclACI`. You can reduce this overhead by moving entry specific directives from `orclACI` to `orclEntryLevelACI`.

The orclEntryLevelACI Attribute for Entry-Level Access Control

When a policy pertains only to a specific entity—for example, a special user—you can maintain, within a single entry, the ACL directives specific to that entry. Oracle Internet Directory enables you to do this through a user-modifiable operational attribute called `orclEntryLevelACI`. The `orclEntryLevelACI` attribute contains ACL directives that apply to only the entry with which it is associated.

Any directory entry can optionally carry a value for this attribute. This is because Oracle Internet Directory extends the abstract class `top` to include `orclEntryLevelACI` as an optional attribute.

The `orclEntryLevelACI` attribute is multi-valued and has a structure similar to that of `orclACI`. The structure definition is provided later in this chapter.

Access Control 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.

To specify access rights for a group of people or entities, you identify them in access control groups. There are two types of access control groups: ACP groups and privilege groups.

ACP groups If an individual is a member of an ACP group, then the directory server simply grants to that individual the privileges associated with that ACP group.

Use ACP groups to resolve access at the level of an ACP. For example, suppose you want to give to several hundred users access to browse an entry. You could assign the browse privilege to each entry individually, but this could require considerable administrative overhead. Moreover, if you later decide to change that privilege, you would have to modify each entry individually. A more efficient solution is to assign the privilege collectively. To do this, you create a group entry, designate it as an ACP group, assign the desired privilege to that group, then assign users as members of that group. If you later change the access rights, you need to do it in one place, for the group, rather than for each individual user. Similarly, you can remove that privilege from multiple users by removing them from the group, rather than having to access multiple individual entries.

ACP groups are associated with the `orclacpgroup` object class.

Privilege groups A privilege group is a higher-level access group. It is similar to an ACP group in that it lists users with similar rights. However, it also provides for additional checking beyond a single ACP, as follows: if an ACP denies access, an attribute in the user's entry tells the directory server whether the user being denied is in any privilege group. If so, then this user has additional rights at a higher administration level, and all higher administration levels in the DIT are checked. If the directory server finds a higher ACP that grants to the privilege group access to the requested object, then it overrides the denials by the subordinate ACP, and grants access to the user.

Normally, you would implement only ACP groups. The additional checking that privilege groups provide can degrade performance. Use privilege groups only when access control at higher levels needs the right to override standard controls at lower levels.

Use privilege groups to grant access to administrators who are not recognized by ACPs lower in the DIT. For example, suppose that the global administrator in a hosted environment must perform operations in a subscriber's subtree. Because the global administrator's identity is not recognized in the subscriber subtree, the directory server, relying only on the ACPs in that subtree, denies the necessary access. However, if the global administrator is a member of a privilege group, then the directory server looks higher in the DIT for an ACP that grants this privilege

group access rights to that subtree. If it finds such an ACP, then the directory server overrides the denials by ACPs in the subscriber's subtree.

Privilege groups are associated with the `orclPrivilegeGroup` object class

Users in Both Types of Groups If a user is a member of both an ACP group and a privilege group, then the directory server performs an evaluation for each type of group. It resolves access rights for the privilege group by looking to ACPs higher in the DIT.

Overview: Granting Access Rights to a Group To grant access rights to a group of users, you:

1. Create a group entry in the usual way.
2. Associate the group entry with either the `orclPrivilegeGroup` object class or the `orclACPgroup` object class.
3. Specify the access policies for that group.
4. Assign members to the group.

How the Directory Server Computes Access Control Group Membership Entries can have either direct memberships in groups, or indirect memberships in other ACP or privilege 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 that level.

Because Oracle Internet Directory evaluates for access control purposes only access control groups, it does not allow setting access policies for other types of groups. When a user binds with a specific distinguished name (DN), Oracle Internet Directory computes the user's direct membership in access control 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 access control groups. This process continues until there are no more nested groups to be evaluated.

Each access control group, nested or otherwise, must be associated with an access control group object class—either `orclACPgroup` or `orclPrivilegeGroup`. Even if a group is a member of an access control group, the directory server does not consider it for access control purposes unless it is associated with an access control group object class. When it has determined the user's membership in access control groups, the directory server uses that information for the lifetime of the session.

Example: Computing Access Control Group Membership For example, consider the following group of entries, each of which, with the exception of group4, is marked as a privilege group (`objectclass:orclprivilegegroup`). You can set access control policies that apply to the members of group1, group2, and group3.

Group 1

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

Group 2

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

Group 3

```
dn: cn=group3, c=us
cn: group3
objectclass: top
objectclass: groupofUniquenames
objectclass: orclprivilegegroup
uniquemember: cn=group2, c=us
uniquemember: cn=group1, c=us
uniquemember: cn=group4, c=us
```

Group 4

```
dn: cn=group4, c=us
cn: group4
objectclass: top
objectclass: groupofUniquenames
uniquemember: cn=john doe, c=uk
uniquemember: cn=jane doe, c=uk
uniquemember cn=anne smith, c=us
```

Group `cn=group3,c=us` contains the following nested groups:

- `cn=group2,c=us`
- `cn=group1,c=us`
- `cn=group4,c=us`

Access control policies for group3 are applicable to members of group3, group1, and group2 because each of them is marked as a privilege group. These same access

control policies are not applicable to the members of group4 because group4 is not marked as a privilege group.

For example, suppose that the user binds to Oracle Internet Directory as a member of group 4 with the DN `cn=john smith,c=uk`. None of the access policies applicable to the members of group3 will apply to this user. This is because his only direct membership is to a non-privilege group. By contrast, if the user were to bind as `cn=john smith,c=us`—that is, as a member of group1 and group2—then his access rights will be governed by access policies set up for members of group1, group2, as well as group3 (in which group1 and group2 are nested). This is because all three groups are associated with the object class `orclPrivilegeGroup`.

Access Control Information Components

Access control information represents the permissions that various entities or subjects have to perform operations on a given object in the directory. Thus, an ACI consists of three components:

- The object to which you are granting access
- The entities or subjects to whom you are granting access
- The kind of access you are granting

Object: To What Are You Granting Access?

The *object* part of the access control directive determines the entries and attributes to which the access control applies. It can be either an entry or an attribute.

Entry objects associated with an ACI are implicitly identified by the entry or the subtree where the ACI itself is defined. Any further qualification of objects at the level of attributes is specified explicitly in the ACL expressions.

In the `orclACI` attribute, the entry DN component of the object of the ACI is implicitly that of all entries within the subtree starting with the ACP as its topmost entry. For example, if `dc=com` is an ACP, then the directory area governed by its ACI is:

```
.*, dc=com.
```

However, since the directory area is implicit, the DN component is neither required nor syntactically allowed.

In the `orclEntryLevelACI` attribute, the entry DN component of the object of the ACL is implicitly that of the entry itself. For example, if `dc=acme,dc=com` has an entry level ACI associated with it, then the entry governed by its ACI is exactly:

`dc=acme,dc=com`. Since it is implicit, the DN component is neither required nor syntactically allowed.

The object portion of the ACL allows entries to be optionally qualified by a filter matching some attribute(s) in the entry:

`filter=(`*`ldapFilter`*`)`

where *`ldapFilter`* is a string representation of an LDAP search filter. The special entry selector `*` is used to specify all entries.

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

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

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

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

Note: Access to the entry itself must be granted or denied by using the special object keyword ENTRY. Note that giving access to an attribute is not enough; access to the entry itself through the ENTRY keyword is necessary.

See Also: Appendix B, "The Access Control Directive Format" for information about the format or syntax of ACIs

Subject: To Whom Are You Granting Access?

This section describes:

- The entity to whom access is granted
- The bind mode, that is, the authentication mode used to verify the identity of that entity
- The added-object-constraint, which limits what kind of objects a user can add below the parent, once access is granted.

Entity Access is granted to entities, not entries. The entity component identifies the entity or entities being granted access.

You can specify entities either directly or indirectly.

Directly specifying an entity—This method involves entering the actual value of the entity—for example group=managers. You can do this by using:

- The wildcard character (*), which matches any entry
- The keyword SELF, which matches the entry protected by the access
- A regular expression, which matches an entry's distinguished name—for example, dn=regex
- The members of a privilege group object: group=dn

Indirectly specifying an entity—This is a dynamic way of specifying entities. It involves specifying a DN-valued attribute that is part of the entry to which you are granting access. There are three types of DN-valued attributes:

- dnattr—Use this attribute to contain the DN of the entity to which you are granting or denying access for this entry.
- groupattr—Use this attribute to contain the DNs of the administrative groups to which you are granting or denying access for this entry.
- guidattr—Use this attribute to contain the global user identifier (orclGUID) of the entry to which you want to grant or deny access for this entry.

For example, suppose you want to specify that Anne Smith's manager can modify the salary attribute in her entry. Instead of specifying the manager DN directly, you specify the DN-valued attribute: dnattr=<manager>. Then, when John Doe seeks to modify Anne's salary attribute, the directory server:

- Looks up the value for her manager attribute and finds it to be John Doe
- Verifies that the bind DN matches the manager attribute
- Assigns to John Doe the appropriate access

Bind Mode The bind mode specifies the method of authentication to be used by the subject. There are four modes:

- `Simple`: Simple password-based authentication
- `SSLNoauth`: For SSL-based clients with either anonymous or simple password based authentication. This method uses only the encryption feature of SSL.
- `SSLOneway`: For SSL-based clients with server authentication with either anonymous or password-based authentication
- `SSLTwoway`: For SSL-based clients with strong authentication through SSL.

Specifying the bind mode is optional. The directory server verifies that the bind mode of the user is compatible with that of the node with which the user is trying to communicate. The bind mode specified on one node must be compatible with that specified on the node with which it is communicating. For example, if you specify `SSLTwoway` authentication on one node, then the other node must also be configured for this type of authentication.

Added-Object-Constraint When a parent entry has *add* access, it can add objects as entries lower in the hierarchy. The added-object-constraint can be used to limit that right by specifying an *ldapfilter*. (See Appendix B, "The Access Control Directive Format" and Appendix G, "The LDAP Filter Definition".)

Operations: What Access Are You Granting?

The kind of access granted can be one of the following:

- None
- Compare / nocompare
- Search / nosearch
- Browse / nobrowse
- Proxy / noproxy
- Read / noread
- Selfwrite / noselfwrite
- Write / nowrite
- Add / noadd
- Delete / nodelete

Note that each access level can be independently granted or denied. The no*xxx* means *xxx* permission is denied.

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

Access Level	Description	Type of Object
Compare	Right to perform compare operation on the attribute value	Attributes
Read	Right to read attribute values. Even if read permission is available for an attribute, it cannot be returned unless there is browse permission on the entry itself.	Attributes
Search	Right to use an attribute in a search filter	Attributes
Selfwrite	Right to add oneself to, delete oneself from, or modify one's own entry in a list of DNs group entry attribute. Use this to allow members to maintain themselves on lists. For example, the following command allows people within a group to add or remove only their own DN from the member attribute: `access to attr=(member) by dnattr=(member) (selfwrite)` The `dnattr` selector indicates that the access applies to entities listed in the member attribute. The `selfwrite` access selector indicates that such members can add or delete only their own DN from the attribute.	Attributes
Write	Right to modify / add / delete the attributes of an entry.	Attributes
None	No access rights. The effect of granting no access rights to a subject-object pair is to make the directory appear to the subject as though the object were not present in the directory.	Both entries and attributes
Add	Right to add entries under a target directory entry	Entries
Proxy	Allows the subject to impersonate another user	Entries
Browse	Permission to return the DNs in the search result. It is equivalent to the list permission in X.500. This permission is also required for a client to use an entry DN as the base DN in an ldapsearch operation.	Entries
Delete	Right to delete the target entry	Entries

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

Note: The default access control policy grants the following to both entries and attributes: 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 Access Control by Using Oracle Directory Manager

You can view and modify access control information within ACPs by using either Oracle Directory Manager or command-line tools. This section explains how to accomplish these tasks by using Oracle Directory Manager.

Note: Immediately after installing Oracle Internet Directory, be sure to reset the default security configuration as described in "Task 3: Reset the Default Security Configuration" on page 3-9

This section contains these topics:

- Configuring Oracle Directory Manager for Access Control Management
- Viewing an ACP by Using Oracle Directory Manager
- Adding an ACP by Using Oracle Directory Manager
- Adding an ACP by Using the ACP Creation Wizard of Oracle Directory Manager
- Modifying an ACP by Using Oracle Directory Manager
- Granting Entry-Level Access by Using Oracle Directory Manager
- Example: Managing ACPs by Using Oracle Directory Manager

See Also: Appendix A, "Syntax for LDIF and Command-Line Tools" for a description of command-line tools

Configuring Oracle Directory Manager for Access Control Management

You can configure how Oracle Directory Manager displays ACPs, and how it performs searches for ACPs.

Configuring the Display of ACPs in Oracle Directory Manager

Oracle Directory Manager enables you to determine whether the navigator pane displays all ACPs automatically or only as the result of a search. If you have a large number of ACPs, you may want to display them only as the result of a search.

To configure the display of ACPs:

1. In the navigator pane, expand Oracle Internet Directory Servers and select the server you want to configure.
2. On the toolbar, click User Preferences. The User Preferences dialog box appears.
3. Select the Configure Access Control Policy Management tab page.
4. Select either:
   - Always display all ACPs
   - Only display ACPs based on search request
5. Click OK.

> **Note:** To effect your changes, you must restart Oracle Directory Manager.

Configuring Searches for ACPs When Using Oracle Directory Manager

For ACP searches, Oracle Directory Manager enables you to specify:

- The root of the search
- The maximum number of entries retrieved
- The time limit of the search
- The search depth

To configure searches for ACP entries:

1. In the navigator pane, expand Oracle Internet Directory Servers and select the *directory_server_instance*.
2. On the toolbar, choose User Preferences. The User Preferences dialog box appears.
3. Select the Configure Entry Management tab.
4. In the field labeled Maximum number of one-level subtree entries, enter the number of entries you want ACP searches to retrieve.

5. In the Search Time Limit field, enter the maximum number of seconds for the duration of the search.
6. Click OK.

Viewing an ACP by Using Oracle Directory Manager

If you configured Oracle Directory Manager always to display ACPs, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-13, then you can locate and view an ACP as follows:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management. All of the defined ACPs appear both below Access Control Management in the navigator pane.
2. In the navigator pane, under Access Control Management, select an ACP to display its information in the right pane.

The three fields in the Access Control Management pane are:

Field	Description
Path to the Subtree Control Point	Contains the path defined by the ACP. If you have navigated down a tree to this point, the path to this point appears in this field. If you are creating a new ACP, you must enter the path to it here.
Structural Access Items (Entry Level Operations)	Lists access to entries. Items listed in the Structural Access Items box identify an entry by the following categories: ■ By Whom: To whom or what you are granting access (the subject) ■ Bind Mode: Whether bind mode (authentication) is used ■ Access rights: Browse, Add, Proxy, and Delete **See Also:** "Task 2: Modify Structural Access Items" on page 13-31 for instructions on how to modify structural access items

Field	Description
Content Access Items (Attribute Level Operations)	Lists access to attributes within the entry or entries identified in the Entry Filter column. Columns in this window include: ■ By Whom: To whom or what you are granting access (the subject) ■ Bind Mode: Whether bind mode (authentication) is used ■ Op: The matching operation to be performed against the attribute. Choices are EQ (=) and NEQ (!=) ■ Attribute: The specific attribute to which access is granted or denied (the object) ■ Access rights: Read, Search, Write, Selfwrite, or Compare access **See Also:** "Task 3: Modify Content Access Items" on page 13-35 for instructions on how to modify content access items.

If you configured Oracle Directory Manager to display ACPs only as the result of a search, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-13, then you can locate and view an ACP as follows:

1. Expand Oracle Internet Directory Servers > *directory_server_instance*, then select Entry Management. Perform a search for the entry designated as an ACP. The search result appears in the Distinguished Name box in the lower half of the right pane.
2. In the Distinguished Name box, double-click the entry. The corresponding Entry dialog box appears.
3. To view subtree access controls for this ACP, select the Subtree Access tab.

   To view entry level access controls for this ACP, select the Local Access tab.

Adding an ACP by Using Oracle Directory Manager

ACPs are entries that contain prescriptive, that is, inheritable, access control information. This information affects the entry itself and all entries below it. You will most likely create ACPs to broadcast large-scale access control throughout a subtree.

Adding an ACP by using Oracle Directory Manager involves three tasks:

- Task 1: Specify the entry that will be the ACP.
- Task 2: Configure structural access items—that is, ACIs that pertain to *entries*.
- Task 3: Configure content access items—that is, ACIs that pertain to *attributes*.

Task 1: Specify the Entry That Will Be the ACP

1. If you configured Oracle Directory Manager always to display ACPs, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-13, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*.

   b. Select Access Control Management, and go to step 2.

   If you configured Oracle Directory Manager to display ACPs only as the result of a search, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-13, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management.

   b. Select a node where you want the ACP to reside. If there are no ACPs yet configured, then you may select ACPs under "DSE Root".

2. On the toolbar, click Create. A New Access Control Point dialog box appears.
3. In the Path to Entry field, enter the distinguished name (DN) of the entry that will be the ACP. You can alternatively find the DN:

   - By clicking Browse to the right of the Path to Entry field.
   - By looking in the navigator pane under Entry Management

Task 2: Configure Structural Access Items

1. To define structural access items, that is, ACIs that pertain to entries, just below the Structural Access Items window, click Create. The Structural Access Item dialog box appears. It has four tabs: Entry Filter, Added Object Filter, By Whom, and Access Rights.
2. If you want all entries below the ACP to be governed by the ACP, then you do not need to enter anything on the Entry Filter tab page; simply proceed to the next step.

In an ACP, the access rights defined apply to the entry and all its subentries unless other filters restrict access further. If appropriate, use the Entry Filters tab page to identify the entries to which you are specifying access.

You might restrict access to an entry based on one or more of that entry's attributes. For example, you might choose to restrict access to all entries in which the title is manager and in which the organization unit is Americas.

To identify an entry to which you are specifying access:

a. From the menu at the left end of the Criteria bar, select an attribute type.

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

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.

3. Select the Added Object Filter tab page.

You can specify ACIs to restrict the kind of entries a user can add. For example, you can specify an ACI in the DSE root entry that allows users to add only

entries with `objectclass=country`. The directory server then verifies that any new entry complies with the constraints in this filter.

To restrict the kind of entries a user can add:

a. From the menu at the left end of the Criteria bar, select an attribute type.

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

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.

4. Select the By Whom tab page.

a. From the Bind Mode list, select the type of authentication 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

Bind Mode	Description
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption / decryption is used.
SSL One Way	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

The bind mode is optional in subject specification. If you do not set an authentication method, any kind of authentication is accepted. The bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

b. Specify the entity or entities to whom you are granting access.

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User's Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User's Unique ID (orclGUID) Is Identified by Attribute	The global user identifier (`orclGUID`) of the entry to which you want to grant or deny access for this entry
When Session User's Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

5. Select the Access Rights tab page.
   a. Specify what kinds of rights are granted:
      * Browse—Allows the subject to see the entry
      * Add—Allows the subject to add other entries below this entry
      * Delete—Allows the subject to delete the entry
      * Proxy—Allows the subject to impersonate another user
   b. Click OK.

Task 3: Configure Content Access Items

1. To define content access items, that is, ACIs that pertain to attributes, just below the Content Access Items window, click Create. The Content Access Item dialog box appears. Each tab page contains items you can modify.
2. If you want all entries below the ACP to be governed by the ACP, then you do not need to enter anything on Entry Filter tab page; simply proceed to the next step.

   In an ACP, the access rights defined apply to the entry and all its subentries unless other filters restrict access further. If appropriate, use the Entry Filters tab page to identify the entries to which you are specifying access.

   You might restrict access to an entry based on one or more of that entry's attributes. For example, you might choose to restrict access to all entries in which the title is manager and in which the organization unit is Americas.

To identify an entry to which you are specifying access:

a. From the menu at the left end of the Criteria bar, select an attribute type.

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

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.

3. Select the By Whom tab page.

a. From the Bind Mode list, select the type of authentication to be used by the subject (that is, the entity that seeks access). There are five bind modes from which to select:

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

Bind Mode	Description
SSL One Way	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

The bind mode is optional in subject specification. If you do not set an authentication method, any kind of authentication is accepted. The bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

b. Specify the entity or entities to whom you are granting access.

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User's Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User's Unique ID (orclGUID) Is Identified by Attribute	The global user identifier (`orclGUID`) of the entry to which you want to grant or deny access for this entry
When Session User's Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

4. Select the Attribute tab page.

a. From the right menu, select the attribute to which you want to grant or deny access.

b. From the left menu, select the matching operation to be performed against the attribute. Choices are EQ (Equal (=)) and NEQ (Not Equal (!=)).

For example, if you select EQ and `cn`, then the access rights you grant apply to the `cn` attribute. If you select NEQ and `cn`, then the access rights you grant do not apply to the `cn` attribute.

5. Select the Access Rights tab page and specify the items as described in Table 13–1.

Table 13–1 Access Rights for Attributes

Access Right	Description
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
Write	Right to modify / add / delete the attributes of an entry.
Selfwrite	Right to add oneself to, delete oneself from, or modify one's own entry in a list of DNs group entry attribute. Use this to allow members to maintain themselves on lists. For example, the following command allows people within a group to add or remove only their own DN from the member attribute: `access to attr=(member) by dnattr=(member) (selfwrite)` The `dnattr` selector indicates that the access applies to entities listed in the member attribute. The `selfwrite` access selector indicates that such members can add or delete only their own DN from the attribute.
Compare	Right to perform compare operation on the attribute value

6. Click OK to close this dialog box and return to the main Oracle Directory Manager dialog box.

Adding an ACP by Using the ACP Creation Wizard of Oracle Directory Manager

The ACP Creation Wizard guides you through the tasks involved in adding an ACP. These tasks are:

- Task 1: Specify the entry that will be the ACP.
- Task 2: Configure structural access items—that is, ACIs that pertain to *entries*.
- Task 3: Configure content access items—that is, ACIs that pertain to *attributes*.

Task 1: Specify the Entry That Will Be the ACP

1. If you configured Oracle Directory Manager always to display ACPs, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-13, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*.

   b. In the navigator pane, select Access Control Management, and go to step 2.

   If you configured Oracle Directory Manager to display ACPs only as the result of a search, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-13, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management.

   b. In the navigator pane, select a node where you want the ACP to reside. If there are no ACPs yet configured, you may select ACPs under "DSE Root".

2. On the toolbar, click Create. A New Access Control Point dialog box appears.

3. In the Path to Entry field, enter the distinguished name (DN) of the entry that will be the ACP. You can alternatively find the DN by looking in the navigator pane under Entry Management or by clicking Browse.

Task 2: Configure Structural Access Items by Using the ACP Creation Wizard

1. To define structural access items, that is, ACIs that pertain to entries, just below the Structural Access Items window, click Create via Wizard. The first Structural Access Item dialog box appears.

In an ACP, the access rights defined apply either to the entry and all its subentries or to a specific entry only. The next sections tell you how to configure an ACP for either option.

If you specify prescriptive structural access items, then all entries below the ACP are governed by that ACP. If you want prescriptive structural access items, then you do not need to enter anything on this first Structural Access Item dialog box.

1. To identify an entry to which you are specifying access:
   a. From the menu at the left end of the Criteria bar, select an attribute type.
   b. From the menu in the middle of the bar, select one of the following filter options:

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

   c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.
   d. Click Next. A second Structural Access Item dialog box prompts you to specify any ACI's to restrict the kind of entries a user can add.
2. You can specify ACIs to restrict the kind of entries a user can add. For example, you can specify an ACI in the DSE root entry that allows users to add only entries with `objectclass=country`. The directory server then verifies that any new entry complies with the constraints in this filter.

   To restrict the kind of entries a user can add:

   a. From the menu at the left end of the Criteria bar, select an attribute type.

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

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.

d. Choose Next. The wizard prompts you to specify the type of authentication, called the bind mode, and the subject to whom you are granting access.

3. The bind mode is optional in subject specification. If you do not set an authentication method, or choose None, any kind of authentication is accepted. The bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

a. To specify the type of authentication, or bind mode, from the Bind Mode list, select the type of authentication to be used by the subject (that is, the entity that seeks access). There are five bind modes from which to select:

b. To specify the entity or entities to whom you are granting access, select one of the following.

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User's Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User's Unique ID (orclGUID) Is Identified by Attribute	The global user identifier (`orclGUID`) of the entry to which you want to grant or deny access for this entry
When Session User's Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

4. Click Next. A Structural Access Item dialog box prompts you for access rights information. Specify what kinds of rights are granted:
   - Browse: Allows the subject to see the entry
   - Add: Allows the subject to add other entries below this entry
   - Delete: Allows the subject to delete the entry
   - Proxy: Allows impersonating an entity without providing its password
5. Click Finish.

Task 3: Configure Content Access Items by Using the ACP Creation Wizard

To define content access items, that is, ACIs that pertain to attributes, just below the Content Access Items window, click Create via Wizard. The first Content Access Item dialog box appears.

If you specify prescriptive content access items, then all entries below the ACP are governed by that ACP. If you want prescriptive content access items, then you do not need to enter anything on this first Content Access Item dialog box.

1. To identify an attribute to which you are specifying access:
   a. From the menu at the left end of the Criteria bar, select an attribute type.
   b. From the menu in the middle of the bar, select one of the following filter options:

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

   c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.
   d. Click Next. A second Content Access Item dialog box prompts you to specify to whom you are granting access.
2. Specify the type of authentication—called bind mode—to be used by the subject (that is, the entity that seeks access).

   The bind mode is optional in subject specification. If you do not set an authentication method, or choose None, any kind of authentication is accepted. The bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

There are five bind modes from which to select:

Bind Mode	Description
None	No authentication
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption / decryption is used.
SSL One Way	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

3. Specify the entity or entities to whom you are granting access.

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User's Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User's Unique ID (orclGUID) Is Identified by Attribute	The global user identifier (`orclGUID`) of the entry to which you want to grant or deny access for this entry
When Session User's Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

4. Click Next. A Content Access Item dialog box prompts you to select an attribute and the matching operation to be performed against it.

5. To select an attribute and the matching operation to be performed against it:

   a. In the Attribute field of the Content Access Item dialog box, from the right list, select the attribute to which you want to grant or deny access.

   b. From the left list, select the matching operation to be performed against the attribute. Choices are EQ (Equal (=)) and NEQ (Not Equal (!=)).

   c. Click Next. A Content Access Item dialog box prompts you to specify access rights.

6. Specify what kinds of rights are granted as described in Table 13–2.

Table 13–2 Access Rights for Attributes

Access Right	Description
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
Write	Right to modify / add / delete the attributes of an entry.
Selfwrite	Right to add oneself to, delete oneself from, or modify one's own entry in a list of DNs group entry attribute. Use this to allow members to maintain themselves on lists. For example, the following command allows people within a group to add or remove only their own DN from the member attribute: `access to attr=(member) by dnattr=(member) (selfwrite)` The `dnattr` selector indicates that the access applies to entities listed in the member attribute. The `selfwrite` access selector indicates that such members can add or delete only their own DN from the attribute.
Compare	Right to perform compare operation on the attribute value

7. Click Finish.

Modifying an ACP by Using Oracle Directory Manager

Modifying ACPs by using Oracle Directory Manager involves three tasks:

- Task 1: Specify the entry that you want to modify.
- Task 2: Modify structural access items—that is, ACIs that pertain to *entries*.
- Task 3: Modify content access items—that is, ACIs that pertain to *attributes*.

Task 1: Specify the Entry That You Want to Modify

1. If you configured Oracle Directory Manager always to display ACPs, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-13, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management. Select Access Control Management. All of the defined Access Control Policy Points (ACPs) appear in a list below Access Control Management in the navigator pane. They also appear in the right pane.

   b. Under Access Control Management, select the ACP you want to modify. The information for that ACP is displayed in the right pane. Alternatively, you can double-click an ACP in the right pane to display the data in a separate dialog box.

   If you configured Oracle Directory Manager to display ACPs only as the result of a search, as described in "Configuring the Display of ACPs in Oracle Directory Manager" on page 13-13, then begin as follows:

   a. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Access Control Management, and select the ACP you want to modify. The information for that ACP is displayed in the right pane.

   b. Click Edit. The Subtree Access Control Point dialog box appears.

Task 2: Modify Structural Access Items

You can add new structural access items, or modify existing ones.

> **See Also:** "Task 2: Configure Structural Access Items" on page 13-16 for instructions about adding structural access items

To modify structural access items:

1. In the Structural Access Items window, select the item you want to modify, and, just below the Structural Access Items window, click Edit. The Structural Access Item dialog box appears.
2. Use the Entry Filters tab page to narrow the set of entries to which you are granting access. If you want all entries below the ACP to be governed by the ACP, proceed to the next step.

   You might choose an entry based on one or more attributes. For example, you might choose to search for all those whose title is secretary, or for all those whose title is manager and whose organization unit is Americas.

   In the Criteria window of the Entry Filters tab page, use the search criteria bar to select an attribute, enter a value for that attribute, and specify a filter for matching the specified attribute with the value you entered. To do this:

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

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

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.

3. Use the Added Object Filter tab page to specify ACIs to restrict the kind of entries a user can add. For example, you can specify an ACI in the DSE root entry that allows users to add only entries with `objectclass=country`. The directory server then verifies that any new entry complies with the constraints in this filter.

To restrict the kind of entries a user can add:

a. From the menu at the left end of the Criteria bar, select an attribute type.

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

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.

4. Use the By Whom tab page to specify the subject of the ACI (that is, the entity that seeks access).

a. Specify the type of authentication, called bind mode to be used by the subject. There are five bind modes from which to select:

Bind Mode	Description
None	No authentication
SSL No Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. In this case, only SSL encryption / decryption is used.
SSL One Way	Only the directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

The bind mode is optional in subject specification. For the directive to be applicable, the bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

b. Specify the entity or entities to whom you are granting access.

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User's Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User's Unique ID (orclGUID) Is Identified by Attribute	The global user identifier (`orclGUID`) of the entry to which you want to grant or deny access for this entry

Entity	Description
When Session User's Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

5. Select the Access Rights tab page.
   a. Determine what kinds of rights are granted: Browse, Add, Delete, or Proxy. If an entry is unspecified, then access is determined at the next highest level in which access is specified.
   b. Click OK.

Task 3: Modify Content Access Items

You can add new content access items, or modify existing ones.

> **See Also:** "Task 3: Configure Content Access Items" on page 13-20 for instructions about adding new content access items

To modify content access items:

1. In the Content Access Items box, select the content access item you want to modify, then, just below the Content Access Item window, click Edit. The Content Access Items dialog box appears. Each tab page contains items you can modify.
2. If you want all entries below the ACP to be governed by the ACP, then you do not need to enter anything on Entry Filter tab page; simply proceed to the next step.

   In an ACP, the access rights defined apply to the entry and all its subentries unless other filters restrict access further. If appropriate, use the Entry Filters tab page to identify the entries to which you are specifying access.

   You might restrict access to an entry based on one or more of that entry's attributes. For example, you might choose to restrict access to all entries in which the title is manager and in which the organization unit is Americas.

   To identify an entry to which you are specifying access:

   a. From the menu at the left end of the Criteria bar, select an attribute type.

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

Filter	Description
Begins With	Searches by using only the first few characters of the attribute value
Ends With	Searches for an entry by using only the last few characters of the specified attribute value
Contains	Searches for an entry in which the attribute you specified includes, but is not necessarily limited to, the value you enter
Exact Match	Searches for an entry whose specified attribute is the same as the value you enter
Greater or Equal	Searches for an entry in which the specified attribute is numerically or alphabetically greater than or equal to the value you enter. An entry is alphabetically greater if it is closer to the beginning of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. An entry is alphabetically less if it is closer to the beginning of the alphabet.
Present	Determines if an entry with the specified attribute is present at that level of the tree. You do not need to enter a value to use this relationship. For example, the phrase `cn Present` retrieves all entries with a `cn` attribute value at that level of the tree.

c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.

3. Select the By Whom tab page.

a. From the Bind Mode list, select the type of authentication 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.

Bind Mode	Description
SSL Two Way	Both client and server authenticate themselves to each other. They do this by sending certificates to each other.
Simple	The client identifies itself to the server by means of a DN and a password which are sent in the clear over the network. The server verifies that the DN and password sent by the client matches the DN and password stored in the directory.

The bind mode is optional in subject specification. If you do not set an authentication method, any kind of authentication is accepted. The bind mode specified on one node should match the bind mode specified on the node with which it is communicating.

b. Specify the entity or entities to whom you are granting access.

Entity	Description
Everyone (*)	All who try to access the entry
A Specific Group	A previously defined group name
A Specific Entry	A previously defined directory entry
A Subtree	An entire subtree in the directory, which you select
When Session User's Distinguished Name (DN) Is Identified by Attribute	Anyone whose DN is an attribute in the entry. For example, you might want to grant read access to a group entry to members of the group.
When Session User's Unique ID (orclGUID) Is Identified by Attribute	The global user identifier (`orclGUID`) of the entry to which you want to grant or deny access for this entry
When Session User's Distinguished Name (DN) Matches the Accessed Entry	Anyone who has correctly logged in as the entry specified

4. Select the Attribute tab page.

a. From the right menu, select the attribute to which you want to grant or deny access.

b. From the left menu, select the matching operation to be performed against the attribute. Choices are EQ (Equal (=)) and NEQ (Not Equal (!=)).

For example, if you select EQ and cn, then the access rights you grant apply to the cn attribute. If you select NEQ and cn, then the access rights you grant do not apply to the cn attribute.

5. Select the Access Rights tab page and specify the items as described in Table 13–1

Table 13–3 Access Rights for Attributes

Access Right	Description
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
Write	Right to modify / add / delete the attributes of an entry.
Selfwrite	Right to add oneself to, delete oneself from, or modify one's own entry in a list of DNs group entry attribute. Use this to allow members to maintain themselves on lists. For example, the following command allows people within a group to add or remove only their own DN from the member attribute: `access to attr=(member) by dnattr=(member) (selfwrite)` The `dnattr` selector indicates that the access applies to entities listed in the member attribute. The `selfwrite` access selector indicates that such members can add or delete only their own DN from the attribute.
Compare	Right to perform compare operation on the attribute value

6. Click OK.

Granting Entry-Level Access by Using Oracle Directory Manager

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

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Entry Management. You may either:
   - Select the entry to display its properties in the right pane
   - Use the search panel to find the entry, then double-click the entry to open the Entry dialog box.
2. Select the Local Access tab page, then create and edit local ACIs in the Structural Access Item and Content Access Item boxes.

3. Once you have made the changes, click Apply.

> **Note:** You must click Apply to send the information you just entered to the directory server. If you do not click Apply, the information you just entered is simply held in the Oracle Directory Manager cache.

Example: Managing ACPs by Using Oracle Directory Manager

This example illustrates how to use Oracle Directory Manager to create a new ACP that has ACIs within it. Suppose you are an administrator in a large company, and you want to limit access to user passwords, so that everyone can compare a password, but only the owner of each password, that is, the user, can read the password or modify it.

In this example, we create a new ACP and populate it with four ACIs that set the following permissions:

- Limited access to a `userpassword` attribute by everyone
- Open access to the same `userpassword` attribute by the user himself
- Open access to all attributes except `userpassword` to everyone
- Open access to all attributes to everyone

Create a New ACP

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

Configure Structural Access Items To set the access rights for an entry:

1. Just below the Structural Access Items box, click Create. A Structural Access Items dialog box appears. It contains three tabs: Entry Filter, By Whom, and Access Rights.

Because you want the ACIs to apply to all entries under the ACP, do not use the Entry Filter tab page.

2. Select the By Whom tab page to define the subject of the ACI. From the Bind Mode list, select the authentication mode appropriate to your environment. To create access rights for everyone, select Everyone.
3. Select the Access Rights tab page. By default, all rights—browse, add, and delete—are granted. Proxy is unspecified.
   a. Change the access rights so that Everyone can browse all entries, but cannot add or delete them.
   b. Click OK.

Configure Content Access Items The four ACIs in this example use the same structural content item information. They differ only in the content access they allow. The rest of this section describes how to create the content access for the ACIs.

To define the content access items:

1. Below the Content Access Items box, click Create. The Content Access Items dialog box appears.

   Because you want this ACI to apply to all entries under the ACP, do not use the Entry Filter tab page.
2. Select the By Whom tab page, select Everyone.
3. Select the Attribute tab page. This page has two fields. The first has two choices: EQ (equals) and NEQ (not equals). The second sets the attribute.

   Select EQ and select `userPassword`.
4. Select the Access Rights tab page. By default, all permissions are granted. Change the permissions so that read, search, write, and compare are denied.
5. Click OK.

   You have completed one ACI.

Create Another ACI Create another ACI that allows a user to read, write, search, and compare his own password.

1. Under the Content Access Items box, click Create. The Content Access Items dialog box appears.
2. Select the By Whom tab page. Click When Session User's Distinguished Name (DN) Matches the Accessed Entry.

3. Select the Attribute tab page. This tab page has two lists.The first has two choices: EQ (equals) and NEQ (not equals). The second sets the attribute.

   Select EQ and userPassword.

4. Select the Access Rights tab page.

   Grant access to read, search, write, and compare. Leave selfwrite unspecified.

5. Click OK.

You have now created two ACIs. One denies Everyone read, search, write, and compare access to the `userPassword` attribute. The second allows the owner of the password to read, search, write, and compare that attribute.

Create a Third ACI

The next ACI grants access to Everyone to read, search, and compare all attributes except `userPassword`. It denies write access.

1. Under the Content Access Items field, click Create to display the Content Access Items.
2. Select the By Whom tab page. Select Everyone.
3. Select the Attribute tab page.

   Select NEQ and `userPassword`.

   This combination means that any attribute that is *not* equal to `userpassword` is the object of the permissions in this ACI.

4. Select the Access Rights tab page.

   Grant access to read, search, and compare. Deny write access. Leave selfwrite unspecified.

5. Click OK to apply these permissions and close the dialog box.

Create a Fourth ACI

The next ACI grants access to Self to read, browse, and write all attributes except `userpassword`. Including this ACI avoids any ambiguity about whether Self has the same access permissions as Everyone to attributes other than `userPassword`.

1. Under the Content Access Items field, click Create to display the Content Access Items dialog box.
2. Select the By Whom tab page.

Click When Session User's Distinguished Name (DN) Matches the Accessed Entry.

3. Select the Attribute tab page.

   From the lists, select NEQ and `userPassword`. This combination means that any attribute that is *not* equal to `userPassword` is the object of the permissions in this ACI.

4. Press the Access Rights tab page.

   Grant access to read, search, and write. Leave Selfwrite unspecified.

5. Click OK to apply these permissions and close the dialog box.

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

Managing Access Control by Using Command-Line Tools

As described in "Overview of Access Control Policy Administration" on page 13-2, directory access control policy information is represented as user-modifiable operational attributes. Hence, you can manage directory access control by using ldapmodify to set and alter values of these attributes. Any tool, including ldapmodify and ldapmodifymt, can be used for this purpose.

To directly edit the ACI, you should understand the format and semantics of the directory representation of the ACI as described in Appendix B, "The Access Control Directive Format".

See Also:

- "LDAP Data Interchange Format (LDIF) Syntax" on page A-2 for information about how to format input by using **LDAP Data Interchange Format (LDIF)**, the required input format for line mode commands
- "ldapmodify Syntax" on page A-15 for information about how to run ldapmodify
- Appendix B, "The Access Control Directive Format" for information about the format or syntax of ACI

Example: Restricting the Kind of Entry a User Can Add

You can specify ACIs to restrict the kind of entries a user can add. For example, you can specify an ACI in the DSE root entry that allows users to add only entries with `objectclass=country`. To do this, you use the `added_object_constraint` filter. The directory server then verifies that any new entry complies with the constraints in this filter.

The following example specifies that:

- The subject `cn=admin,c=us` can browse, add, and delete under `organization` entries.
- The subject `cn=admin,c=us` can add `organizationalUnit` objects under `organization` entries
- All others can browse under `organization` entries

```
access to entry filter=(objectclass=organization)
by group="cn=admin,c=us"
          constraintonaddedobject=(objectclass=organisationalunit)
          (browse,add,delete)
by * (browse)
```

Example: Setting Up an Inheritable ACP by Using ldapmodify

This example sets up subtree access permissions in an `orclACI` at the **root DSE** by using an LDIF file named `my_ldif_file`. Because this example refers to the `orclACI` attribute, this access directive governs all the entries in the DIT.

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

The LDIF file, `my_ldif_file`, contains the following:

```
dn:
changetype: modify
replace: orclaci
orclaci: access to entry
    by dn="cn=directory manager, o=IMC, c=us" (browse, add, delete)
    by * (browse, noadd, nodelete)
orclaci: access to attr=(*)
    by dn="cn=directory manager, o=IMC, c=us" (search, read, write, compare)
    by self (search, read, write, compare)
    by * (search, read, nowrite, nocompare)
```

Example: Setting Up Entry-Level ACIs by Using ldapmodify

This example sets up entry-level access permissions in the `orclEntryLevelACI` attribute by using an LDIF file named `my_ldif_file`. Because this example refers to the `orclentrylevelACI` attribute, this access directive governs only the entry in which it resides.

```
ldapmodify -v -h myhost -D "cn=Directory Manager, o=IMC, c=US" -w "controller"
-f my_ldif_file
```

The LDIF file, `my_ldif_file`, contains the following:

```
dn:
changetype: modify
replace: orclentrylevelaci
orclentrylevelaci: access to entry
    by dn="cn=directory manager, o=IMC, c=us" (browse, add, delete)
    by * (browse, noadd, nodelete)
orclentrylevelaci: access to attr=(*)
    by dn="cn=directory manager, o=IMC, c=us" (search, read, write, compare)
    by * (search, read, nowrite, nocompare)
```

Note: In this example, no DN value is specified. This means that this ACI pertains to the root DSE and its attributes only.

Example: Using Wild Cards

This example shows the use of wild cards (*) in the object and subject specifiers. For all entries within the `acme.com` domain, it grants to everyone browse permission on all entries, as well as read and search permissions on all attributes.

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

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

Note that, in order to allow reading the attributes, browse permissions must be granted on the entries in order for read permissions to be granted to the attributes of those entries.

Example: Selecting Entries by DN

This example shows the use of a regular expression to select the entries by DN in two access directives. It grants to everyone read-only access to the address book attributes under dc=acme,dc=com access.

orclACI attribute of dc=acme, dc=com:

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

orclACI attribute of dc=us, dc=acme, dc=com:

```
access to entry by * (browse)
access to attr=(*) by dn=".*,dc=us,dc=acme,dc=com" (search, read)
```

Example: Using Attribute and Subject Selectors

This example shows the use of an attribute selector to grant access to a specific attribute, and various subject selectors. The example applies to entries in the dc=us,dc=acme,dc=com subtree. The policy enforced by this ACI can be described as follows:

- For all entries within the subtree, the administrator has add, delete, and browse permissions. Others within the dc=us subtree can browse, but those outside it have no access to the subtree.
- The salary attribute can be modified by one's manager and viewed by oneself. No one else has access to the salary attribute.
- The userPassword attribute can be viewed and modified by oneself and the administrator. Others can only compare this attribute.
- The homePhone attribute can be read and written by oneself and viewed by anyone else.
- For all other attributes, only the administrator can modify values. Everyone else can compare, search, read, but cannot update attribute values.

"orclACI" attribute of "dc=us, dc=acme, dc=com":

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

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


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


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


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

Example: Granting Read-Only Access

This example gives to everyone read-only access to address book attributes under `dc=acme,dc=com`. It also extends to everyone read access to all attributes within the `dc=us,dc=acme,dc=com` subtree only.

`orclACI` attribute of `dc=acme, dc=com`:

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

`orclACI` attribute of `dc=us, dc=acme, dc=com`:

```
access to entry by * (browse)
access to attr=(*) by dn=".*,dc=us,dc=acme,dc=com" (search, read)
```

Example: Granting Selfwrite Access to Group Entries

This example allows people within the US domain to add or remove only their own name (DN) to or from the member attribute of a particular group entry, for example, a mailing list.

`orclEntryLevelACI` attribute of the group entry in question:

```
access to attr=(member)
by dn=".*, dc=us,dc=acme,dc=com" (selfwrite)
```

How ACL Evaluation Works

When a user tries to perform an operation on a given object, the directory server determines whether that user has the appropriate access to perform that operation on that object. If the object is an entry, it evaluates the access systematically for the entry and each of its attributes.

Evaluating access to an object—including an attribute of an entry—can involve examining all the ACI directives for that object. This is because of the hierarchical nature of ACPs and the inheritance of policies from superior ACPs to subordinate ACPs.

The directory server first examines the ACI directives in the entry-level ACI, `orclEntryLevelACI`. It proceeds to the nearest ACP, then considers each superior ACP in succession until the evaluation is complete.

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.

In all operations except search, the evaluation stops if:

- Access to the entry itself is denied
- Any of the attributes reach the resolved with denial state.

In this case the operation would fail and the directory server would return an error to the client.

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

ACL Evaluation Precedence Rules

An LDAP operation requires the BindDN, or subject, of the LDAP session to have certain permissions to perform operations on the objects—including the entry itself and the individual attributes of the entry.

Typically, there could be a hierarchy of access control administration authorities, starting from the root of a naming context down to successive administrative points (or access control policy points). An ACP is any entry which has a defined value for the `orclACI` attribute. Additionally, the access information specific to a single entry can also be represented within the entry itself (`orclEntryLevelACI`).

ACL evaluation involves determining whether a subject has sufficient permissions to perform an LDAP operation. Typically an `orclentryLevelACI` or `orclACI` might not contain all the necessary information for ACL evaluation. Hence, all available ACL information is processed in a certain order until the evaluation is fully resolved.

That order of processing follows these rules:

- The entry level ACI is examined first. ACIs in the `orclACI` are examined starting with the ACP closest to the target entry and then its superior ACP and so on.
- At any point, if all the necessary permissions have been determined, the evaluation stops; otherwise, the evaluation continues.
- Within a single ACI, if the entity associated with the session DN matches more than one item identified in the *by* clause, the effective access evaluates to:
  - The union of all the granted permissions in the matching by clause items

    ANDed with
  - The union of all the denied permissions in the matching by clause items

Precedence at the Entry Level ACIs at the entry level are evaluated in the following order:

1. With a filter. For example:

   ```
   access to entry filter=(cn=p*)
       by group1 (browse, add, delete)
   ```

2. Without a filter. For example:

   ```
   access to entry
       by group1 (browse, add, delete)
   ```

Precedence at the Attribute Level At the attribute level, specified ACIs have precedence over unspecified ACIs.

1. ACIs for specified attributes are evaluated in the following order:
   a. Those with a filter. For example:

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

   b. Those without a filter. For example:

      ```
      access to attr=(salary)
      by group1 (search, read)
      ```

2. ACIs for unspecified attributes are evaluated in the following order:
   a. With a filter. For example:

      ```
      access to attr=(*) filter (cn=p*)
      by group1 (read, write)
      ```

   b. Without a filter. For example:

      ```
      access to attr=(*)
      by group1 (read, write)
      ```

More Than One ACI for the Same Object

If there are two or more ACIs at the same ACP for the same object, then only one ACI is checked, and all other ACIs are ignored. For example, suppose you have the following two ACIs at the same ACP for the same entry:

- ACI #1:

  ```
  access to entry
         by dn="cn=admin, dc=us,dc=acme,dc=com" (browse, add, delete)
  ```

- ACI #2:

  ```
  access to entry
         by dn="cn=manager,dc=us,dc=acme,dc=com" (search, read)
  ```

If ACI #2 happens to be checked first, then the access granted specifically to the administrator in ACI #1 is ignored. If an administrator then seeks access to the entry, then that access is not be resolved at this level of the hierarchy. The evaluation must 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, then the access granted to self in ACI #2 is ignored. If a user then wishes to change his or her own password, then 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, you can specify access to all other objects except that one. You do this either by granting access to all the objects, or by denying access to the one object.

In the following example, access is granted to all attributes:

```
access to attr=(*)
by group2 (read)
```

In the following example, access is denied to the `userpassword` attribute:

```
access to attr!=(userpassword)
by group2 (read)
```

ACL Evaluation For Groups

If an operation on an attribute or the entry itself is explicitly denied at an ACP low in the DIT, then, typically, the ACL evaluation for the attribute (or entry) is

considered "Resolved with Denial." However, if the user of the session (bindDN) is a member of a group object, then the evaluation continues as if it is still unresolved. If permissions are granted to the user of the session at an ACP higher in the tree through a group subject selector, then such grants have precedence over any denials lower in the tree.

This scenario is the only case in which an ACL policy at a higher level ACP has precedence over an ACP policy lower in the DIT.

Access Level Requirements for LDAP Operations

The following table lists LDAP operations and the access required to perform each one.

Operation	Required Access
Create an object	Add access to the parent entry
Modify	Write access to the attributes that are being modified
ModifyDN	Delete access to the current parent and Add access to the new parent.
ModifyDN (RDN)	Write access to the naming attribute, that is, the RDN attribute
Remove an object	Delete access to the object being removed
Compare	Compare access to the attribute
Search	■ Search access on the filter attributes and browse access on the entry (if only the entry DN needs to be returned as a result) ■ Search access on the filter attributes, browse access on the entry, and read permission on the attributes (for all attributes whose values need to be returned as a result)

Part IV

Directory Deployment

This part discusses important deployment considerations. It includes these chapters:

14

General Deployment Considerations

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

This chapter contains these topics:

See Also:

- Chapter 19, "Capacity Planning Considerations" for more detailed information about capacity planning
- Chapter 21, "High Availability And Failover Considerations" for more detailed information about high availability
- Chapter 20, "Tuning Considerations" for more detailed information about tuning
- Part VI: "The Directory and Clusters" for information about failover in clustered environments

The Expanding Role of Directories

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

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

Enterprises should plan a robust and flexible deployment to accommodate:

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

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

Logical Organization Of Directory Information

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

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

This section contains these topics:

Directory Entry Naming

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

DIT Hierarchy and Structure

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

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

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

Physical Distribution: Partitions and Replicas

You can distribute directory data in two ways:

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

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

This section contains these topics:

An Ideal Deployment

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

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

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

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

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

- Establish a network of two or more directory nodes, each holding all the naming contexts. Set up these nodes in a multimaster configuration.
- Deploy these individual nodes, one in each geographic region, to suit the corporate data network connectivity. For example, if a region is connected to the rest of the network by way of a slow link, then it is better to locate a dedicated directory server for use by the clients in that region.
- Individually configure each regional server for failover and recovery.

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

See Also: "Failover Considerations" on page 14-7 for a discussion of redundancy

Partitioning Considerations

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

Typically, the reasons for maintaining partitions are:

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

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

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

Replication Considerations

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

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

- Local accessibility and performance requirements

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

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

- Load balancing

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

- Failure tolerance and higher overall system availability

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

Failover Considerations

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

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

- Intelligent Client Failover

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

- Intelligent Network Level Failover

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

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

See Also:

- Chapter 21, "High Availability And Failover Considerations" for further details about high-availability and failover options available with Oracle Internet Directory
- Part VI: "The Directory and Clusters" for information about failover in clustered environments

About Capacity Planning, Sizing, and Tuning

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

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

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

- Planning phase

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

- Implementation phase

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

This section contains these topics:

Capacity Planning

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

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

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

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

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

Sizing Considerations

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

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

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

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

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

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

The data in this table makes the following assumptions:

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

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

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

See Also: Chapter 19, "Capacity Planning Considerations."

Tuning Considerations

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

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

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

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

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

- CPU usage

  This is determined, to a large extent, by:

  – The number of Oracle directory servers

  – The number of database connections opened by each server

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

- Memory usage

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

- Disk usage

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

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

  See Also: Chapter 20, "Tuning Considerations" for further details on various tuning tips and techniques

Running Multiple Installations of Oracle Internet Directory on One Host

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

See Also: "Identifying a Node as Independent of Its Host" on page 23-31 to configure replication between two Oracle Internet Directory installations on the same host

15

Oracle Components and Oracle Internet Directory

Many Oracle components use Oracle Internet Directory for a variety of purposes. In doing this, they rely on a consolidated Oracle Internet Directory schema and a default Directory Information Tree (DIT). This chapter:

- Describes the consolidated Oracle Internet Directory schema used by various components
- Describes a default DIT structure available when using the various Oracle components
- Describes how you can modify the default DIT structure to accommodate the needs of hosted application environments

This chapter contains these topics:

About Oracle Components and Directory Usage

Oracle Internet Directory enables Oracle components to:

- Maintain for each user a single, global identity across the application environment
- Store and administer configuration information for components in a central place

This chapter considers two general types of environment:

- Hosted—In a hosted environment, one enterprise—for example, an application service provider—makes Oracle components available to multiple other enterprises and stores information for them. In such hosted environments, the enterprise performing the hosting is called the default subscriber, and the enterprises that are hosted are called subscribers. A global administrator performs activities that span the entire directory. Other administrators may be delegated to exercise roles in specific subscriber domains, or for specific applications.
- Non-hosted—In a non-hosted environment, in which there are no subscribers, the enterprise installing Oracle Internet Directory for use with Oracle components is called the default subscriber.

Directory schema and DIT requirements are defined with enough flexibility to accommodate both deployment models.

Ready-to-Use Default Configuration

To make it easy for you to start using Oracle components that use the directory, Oracle Universal Installer creates a default schema and directory information tree (DIT) during Oracle Internet Directory installation. This default DIT framework is the same for both hosted and non-hosted environments. It is flexible; you can modify it to suit the needs of your deployment.

During Oracle Internet Directory installation, the Oracle Universal Installer creates:

- Base schema elements—that is, attributes and object classes—some of which are defined by the Internet Engineering Task Force (IETF), and some of which are specific to Oracle components
- The root Oracle context, a directory container of information common to all Oracle components in the entire site

- Default subscriber Oracle context containing information common to all Oracle components in the subscriber's subtree
- Subscriber Oracle context containing information common to all Oracle components in the subscriber's subtree
- A default password policy for each subscriber. This password policy will then apply to all users in the subscriber user base.

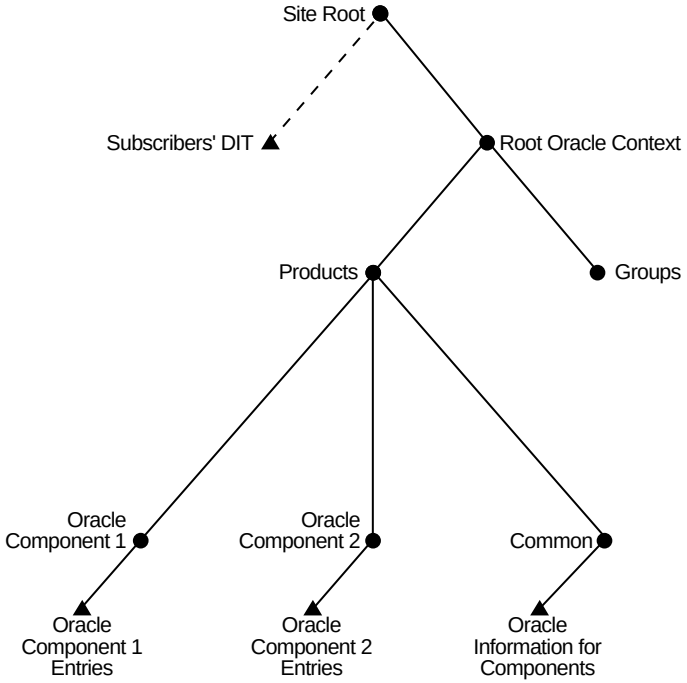
The Root Oracle Context

The root Oracle context includes:

- Site-wide information, including such metadata as default parameter settings, default profiles, and default authorization policies
- A discovery mechanism for Oracle components in a hosted environment to find necessary information for each subscriber

Figure 15–1 shows the organization of the root Oracle context.

Figure 15–1 The Root Oracle Context



Some of the discovery-related information stored at the root Oracle context includes:

- Subscriber Search Base (`orclSubscriberSearchBase`)

  This attribute identifies the node in the DIT under which all the subscribers are placed. This attribute becomes particularly important in the hosted scenario because it provides a common point for all the products to locate a subscriber. For example, in Figure 15–1, `Subscriber` serves as the search base for locating a subscriber. In a non-hosted environment, the value of this attribute points to the parent of the default subscriber.

- Subscriber Nickname Attribute (`orclSubscriberNickNameAttribute`)

  This attribute identifies the nickname attribute to be used when searching for a subscriber under the subscriber search base. For example, because a subscriber is typically represented as an organization, the attribute o can be used as the nickname attribute.

- Default Subscriber (`orclDefaultSubscriber`)

This attribute points to the default subscriber node in the DIT.

In both hosted and non-hosted scenarios, a component finds the correct node in the DIT by using the `orclSubscriberSearchBase` and `orclSubscriberNickNameAttribute` attributes. Once the component finds the appropriate subtree, it obtains the subscriber-specific information it needs from the Oracle context in that subtree.

For example, Oracle9*i*AS Single Sign-On uses this framework for authenticating a user in a hosted scenario. When a user logs in, Oracle9*i*AS Single Sign-On prompts the user for a subscriber. Then, when it looks for an entry, the Oracle9*i*AS Single Sign-On server finds the correct subscriber node in the DIT by using the `orclSubscriberSearchBase` & `orclSubscriberNickName` attributes. Once it learns where the subscriber-specific information resides, it then looks in the subscriber-specific Oracle context to find the location of the user.

If a client does not specify a subscriber, then Oracle Internet Directory assumes that the user is looking for information in the default subscriber subtree.

The Subscriber Oracle Context

A subscriber-specific Oracle context includes:

- A discovery mechanism for Oracle components to find necessary information in the subscriber's subtree
- Oracle component data specific to a subscriber
- An access policy at the subscriber node to protect subscriber data from other subscribers
- A default password policy applicable to all users is placed in the Common container. The attribute `orclCommonUserSearchBase` must be set to the appropriate value for the password policy to be enforced, i.e., the policy will be applied to a user whenever the corresponding attribute matches the value of `orclCommonUserSearchBase`.

Figure 15–2 shows the organization of a subscriber-specific Oracle context.

Figure 15–2 Subscriber-Specific Oracle Context

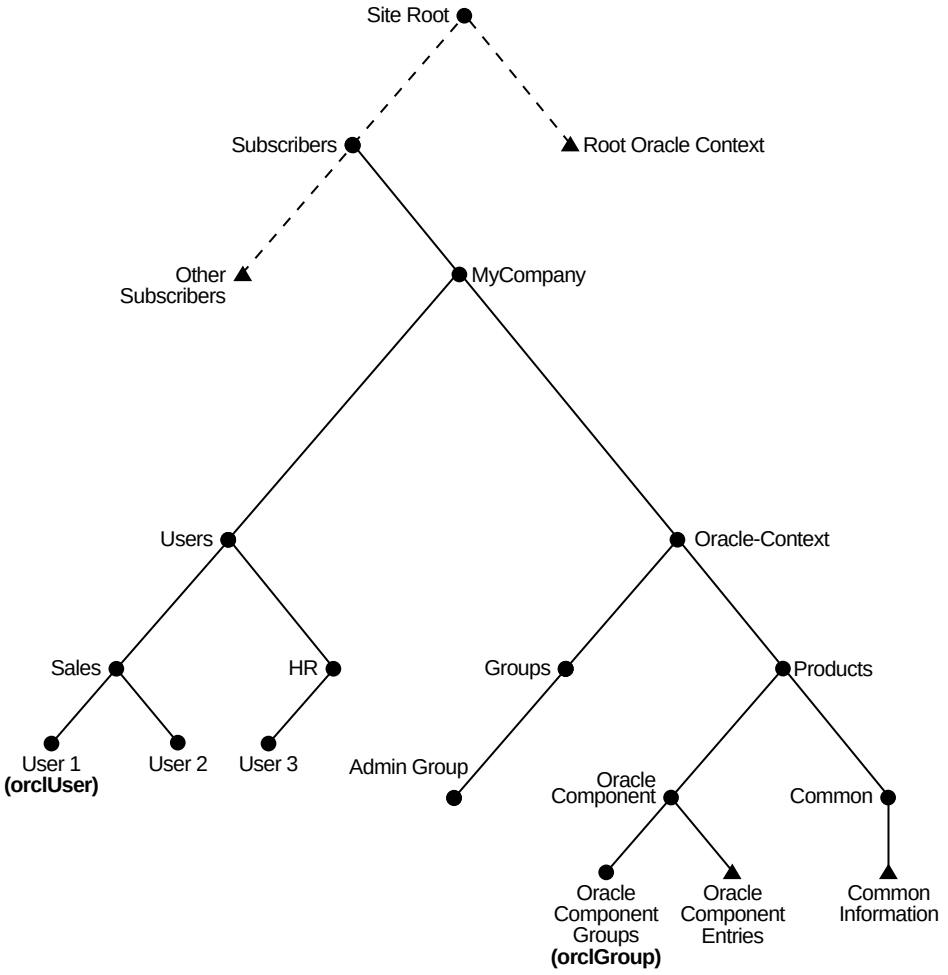


Figure 15–2 shows subscriber-wide information in the directory for an Oracle component and information common to all components. It illustrates two aspects:

- An example of the default subscriber is shown under the container called "Users."
- An example of the default user is shown under the container called "Oracle Context."

The Common entry in the subscriber-specific Oracle context contains information for locating users and groups. Specifically, it includes:

- User Search Base (`orclCommonUserSearchBase`)

  This attribute specifies the node in the subscriber DIT under which all the users are placed. For example, in Figure 15–2 on page 15-6, `users` serves as the search base while searching for a user in a subscriber.

- User Nickname attribute (`orclCommonUserNickNameAttribute`)

  This attribute specifies the nickname attribute to be used when searching for a user under the user search base. For example, when a user logs in, Oracle9*i*AS Single Sign-On prompts the user for the value of this attribute.

- Group Search Base (`orclCommonGroupSearchBase`)

  This attribute specifies the node in the subscriber DIT under which all the groups can be found.

- Object classes to be used for creating user entries (`orclUserObjectClass`)

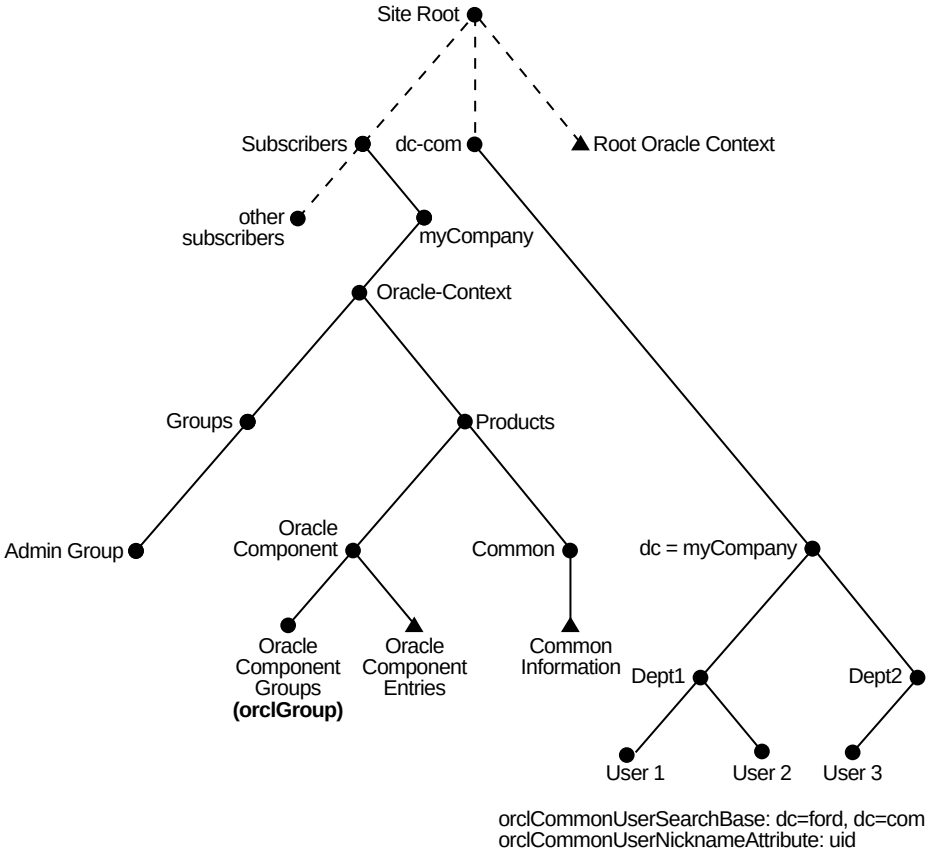
  This attribute specifies a list of object classes to be used when creating user entries under the subscriber tree—for example, `person`, `organizationalPerson`, `inetOrgPerson`, `orclUser`, and so on. For example, the Delegated Administration Service uses this attribute in configuring users.

In a hosted scenario, you might dedicate a particular instance of a component to multiple subscribers. For example, each subscriber might have its own instance of the Oracle9*i*AS Portal component. In this case, the instance information and other data required by each individual subscriber is stored in each subscriber's Oracle context. General information required by all subscribers is stored in the root Oracle context.

In Figure 15–2 on page 15-6, the dotted line between the user and the subscriber shows some of the flexibility with which you can organize a subscriber subtree. You can create and store user data in different ways—for example, you can store it:

- Directly under the subscriber node
- Outside the subscriber tree as illustrated in Figure 15–3

Figure 15–3 Separation of a Subscriber and Subscriber's User Information



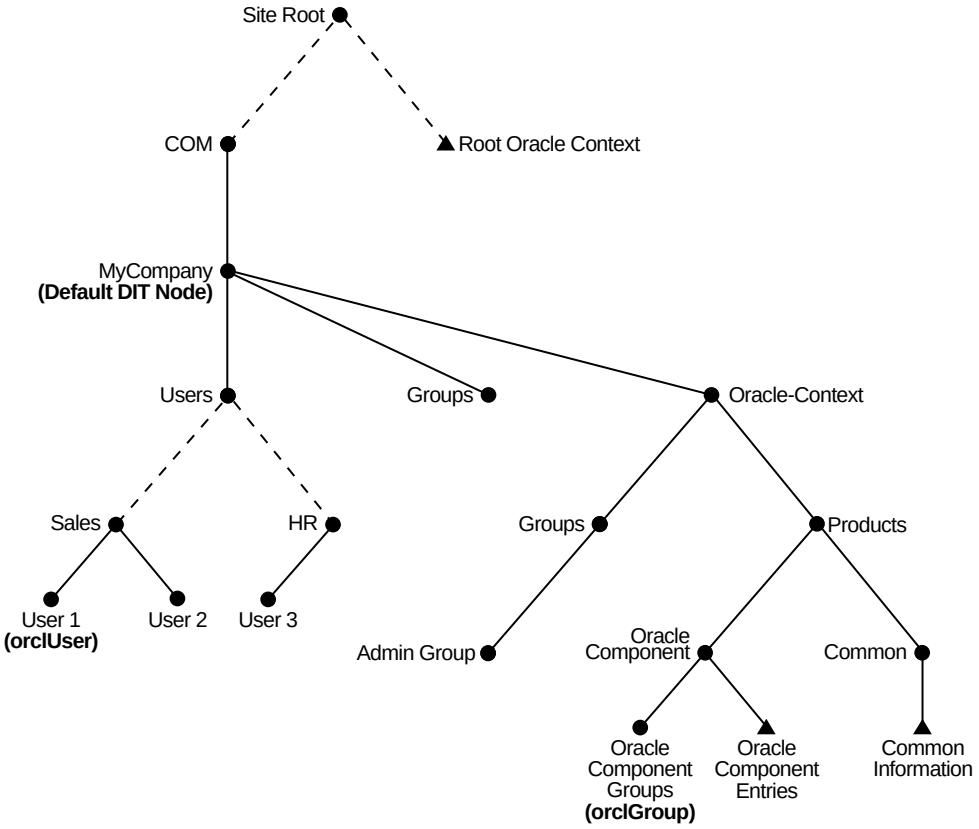
As Figure 15–3 shows, you are not required to create a subscriber's users under the subscriber node itself. The `orclCommonUserSearchBase` attribute in the Common entry for each subscriber-specific Oracle context points to the node containing the user data—in Figure 15–3, it is `dc=myCompany,dc=com`. This

enables subscribers to keep the DNs they may already have, without having to migrate them to a different DIT structure.

A Default Subscriber Configuration

Figure 15–4 shows the DIT for a default subscriber in a non-hosted environment.

Figure 15–4 Default DIT in Non-Hosted Environment



During an Oracle Internet Directory installation, Oracle Universal Installer determines the domain information for the site where it is installing Oracle Internet Directory. It establishes the default DIT structure based on this information. For

example, if Oracle Internet Directory is installed at `My_Company.com`, then Oracle Universal Installer creates the following nodes in the DIT:

- The root Oracle context containing information common to all Oracle components
- The node marked as `Com` in the above figure
- The My_Company node and, under it, the Oracle context
- `User` and `Group` containers under the default subscriber node—in this example, My_Company.com

If you use the default DIT for your enterprise, then you do not need to configure anything at the root Oracle context. Instead, depending on the structure of the subtree that your deployment uses, you simply do the following:

- Configure the user search base and related discovery information in the Oracle context in the default subscriber node. For example, in the deployment in Figure 15–4 on page 15-9, the user search base and related discovery information reside in the `Common` container under `cn=Products,cn=Oracle Context,o=GM`.
- Place user entries in the `Users` container, and group entries in the `Groups` container, both of which reside immediately below the default subscriber node.

In a hosted environment, you would create subscribers at the same level in the DIT as the default subscriber node itself.

As part of Default DIT Creation a seed user is also created to help bootstrap using the Delegated Administration Service and other tools. The user is identified by the following DN: `cn=orclAdmin,cn=users,cn=my_company, dc=com`. The initial password for the user is the same as the Oracle Internet Directory super user (`cn=orcladmin`) password. By default, this user is allowed to create, delete, and edit users under the `cn=Users` container or create, delete, and edit groups under the `cn=Groups` container.

The user also has permission to change the Delegated Administration Service configuration in Oracle Internet Directory. By using this seed user identity, the administrator can use the Delegated Administration Service to create users and groups, and thereby bootstrap the entire directory environment.

See Also: Chapter 9, "The Delegated Administration Service"

16

Directory-Based Application Security

This chapter discusses how you can exploit the way Oracle Internet Directory stores access control policies to secure applications in a large enterprise and in hosted environments. This chapter contains these topics:

Delegated Directory Administration

Because Oracle Internet Directory stores access control policies as LDAP attributes, you can set metapolicies controlling who can modify them. This enables a global administrator to assign privileges to administrators of specific subtrees—for example, to administrators of applications in a hosted environment. Similarly, a global administrator can delegate to departmental administrators access to the metadata of applications in their departments. Department administrators can then control access to their department applications.

Thus, you can implement access control on two levels:

- Authorization of users

  In this case, the directory stores access control policies that external applications then read and enforce. When a user tries to perform an operation by using an application, the application verifies that the user has the correct authorization to perform the operation.

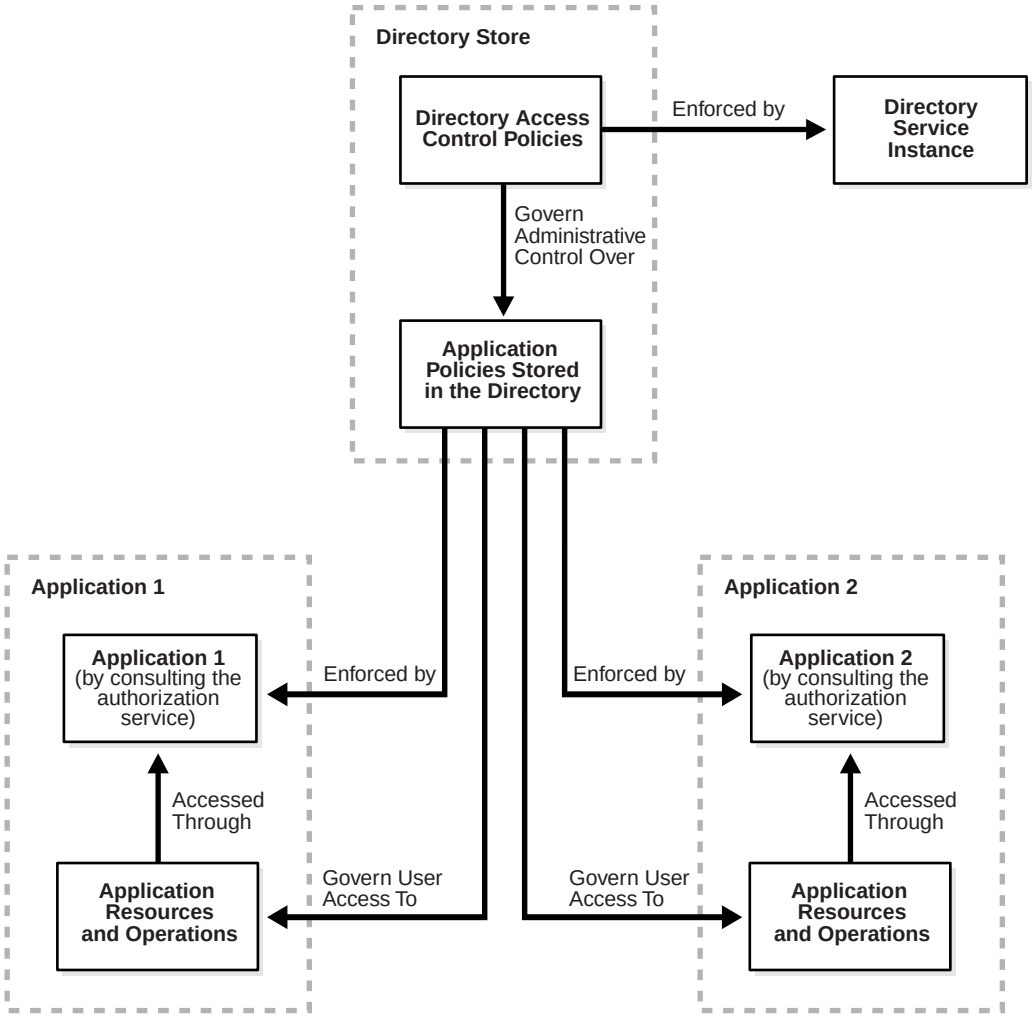
- Authorization of administrators

  In this case, the directory serves as the trusted point of administration for all application-specific access control polices. To govern who can administer the access control policies of specific applications, you set access control policies at the directory level for these applications. Then, when a user attempts to change an application-specific access control policy, the directory verifies that the user has the correct authorization to make that change.

Application-Specific Access Control

Figure 16–1 shows the relationship between directory access control and the application-specific access control mechanisms in a hosted environment.

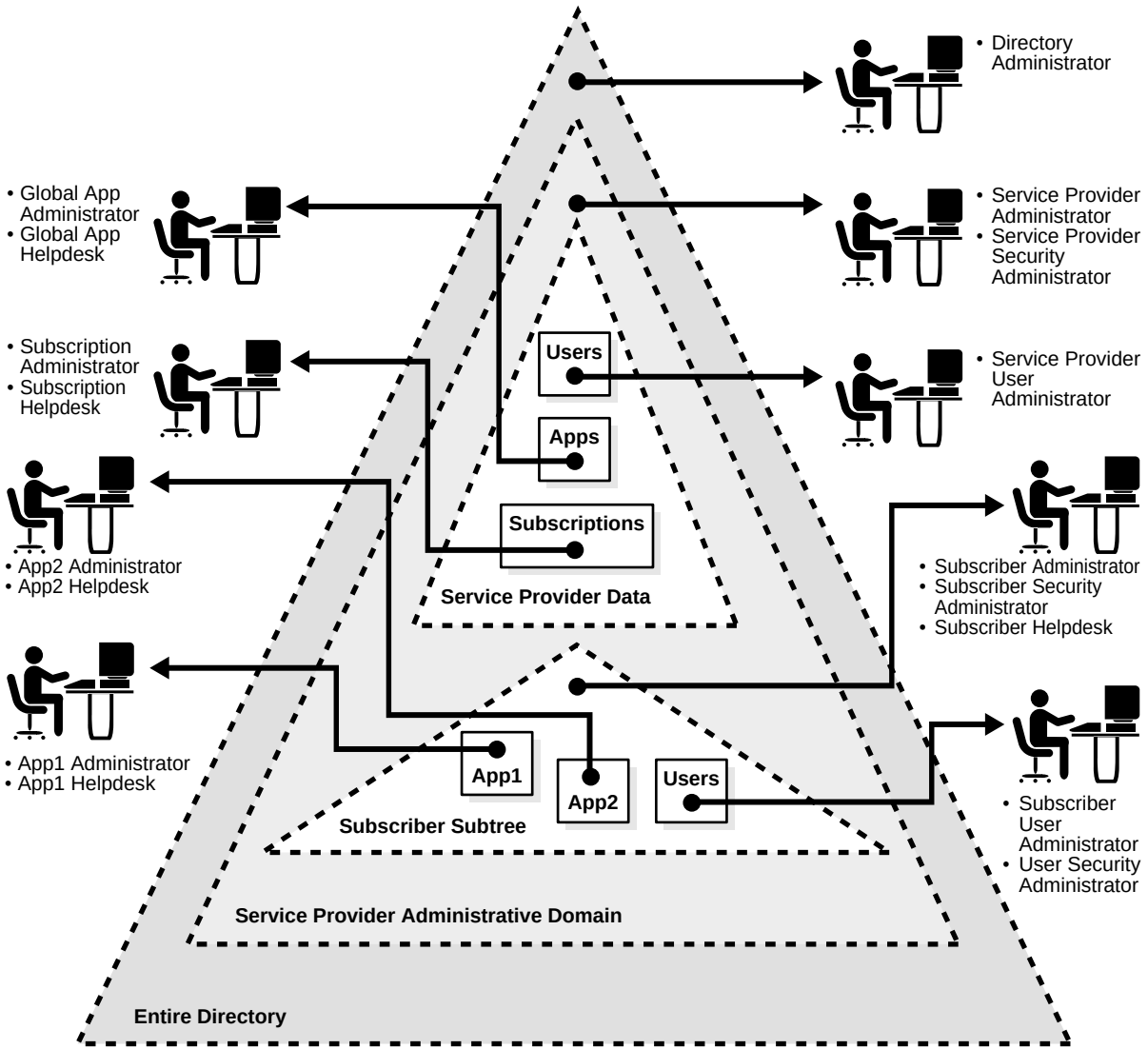
Figure 16–1 Directory Access Control and Application-Specific Access Control



Directory Domains and Roles

Figure 16–2 illustrates the various domains and the roles associated with them in the directory.

Figure 16–2 Directory Domains and Roles in a Hosted Environment



In Figure 16–2, each triangle represents a portion of a DIT.

- The outermost triangle represents the entire directory. The directory administrator has privileges extending across the entire directory.
- Immediately inside the outermost triangle, another triangle represents the service provider administrative domain. In this domain, privileges to add new entries are delegated to the service provider's administrators.
- Inside the service provider administrative domain, privileges can be further delegated based on the ownership of directory information. For example, the delegation can depend on whether the information is private to a specific subscriber or global to the service provider.

Figure 16–2 shows only a single subscriber represented in the directory. In reality there are multiple subscribers, each with its own domain requiring protection from the others.

Some of the protection domains in this model are:

- Entire directory
- Service provider administrative domain
- Service provider-specific directory information tree
- Subscriber-specific subtree
- Application-specific footprint in the directory
- User-specific information

These protection domains are supported by the following roles, which enable the service provider or subscriber to customize access control.

- Global Administrative Roles

  These roles have rights to perform activities that span the entire directory.
- Subscriber-Specific Roles

  These roles are limited to the directory trees specific to the subscribers.
- Application-Specific Roles

When hosting directory-enabled applications, it is not necessary to represent all application-specific roles in the directory. However, it is better that applications, when representing roles that directly affect their directory footprint, follow the delegation model recommendations described earlier. This enables applications to leverage the directory-based delegation model when granting directory-specific privileges to users.

17

Directory Storage of User Authentication Credentials

This chapter explains how Oracle Internet Directory centrally stores security credentials for easy administration by end users and administrators.

This chapter contains these topics:

About Centralized Storage of User Authentication Credentials

Oracle Internet Directory centrally stores security credentials as directory data to make their administration easy for both end users and administrators. When a user leaves a company or changes jobs, that user's privileges should change the same day to guard against misuse of old or unused accounts and privileges. In large enterprises, with user accounts and passwords distributed over multiple databases, an administrator may not be able make all the changes as quickly as good security requires without centralized password administration.

Oracle Internet Directory stores:

- Passwords for authenticating users to the directory itself
- Password verifiers for authenticating users to other Oracle components

Users can store non-Oracle authentication credentials if the non-Oracle applications are directory enabled. These applications must create their own container under the Products entry.

Storing Password Verifiers for Authenticating to Oracle Internet Directory

Oracle Internet Directory stores a user's directory password in the `userPassword` attribute. You can protect this password by storing it as a base 64 encoded string of a one-way hashed value using one of Oracle Internet Directory's supported hashing algorithms. Storing passwords as one-way hashed values—rather than as encrypted values—more fully secures them because a malicious user can neither read nor decrypt them.

During authentication to a directory server, clients supply a password to the directory server in clear text. The directory server hashes this password by using the hashing algorithm specified in the root **directory-specific entry (DSE)** attribute `orclCryptoScheme`. It then verifies it against the hashed password stored in the binding entry's `userPassword` attribute. If the hashed password values match, then the server authenticates the user. If they do not match, then the server sends the user an "Invalid Credentials" error message.

During installation, Oracle Universal Installer prompts you to set the one-way hashing scheme for protecting users' passwords to the directory. It presents you with these options:

- **MD4** —A one-way hash function that produces a 128-bit hash, or message digest

- **MD5**—An improved, and more complex, version of MD4
- **SHA**—Secure Hash Algorithm, which produces a 160-bit hash, longer than MD5. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.
- **UNIX Crypt**—The UNIX hashing algorithm

The hashing algorithm value you specify at installation is stored in the `orclCryptoScheme` attribute in the **root DSE**. You can change that value by using either Oracle Directory Manager or ldapmodify.

Managing Password Protection by Using Oracle Directory Manager

You must be a super user to do the following.

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

1. In the navigator pane, expand Oracle Internet Directory Servers and select the directory server instance for which you want to reset password hashing. The corresponding tab pages for that directory server appear in the right pane.
2. In the System Operational Attributes tab page, in the Password Encryption field, select the type of password hashing you want to use. Options are:
   - MD4
   - MD5
   - SHA
   - UNIX Crypt
   - None. This option specifies that user passwords are stored in clear text.
3. Click Apply.

Managing Password Protection by Using ldapmodify

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

```
ldapmodify -D cn=orcladmin -w welcome -h myhost -p 389 -v -f my_ldif_file
```

The LDIF file, `my_ldif_file`, contains:

```
dn:
changetype: modify
```

```
replace: orclcryptoscheme
orclcryptoscheme: SHA
```

See Also: "Protection of User Passwords for Directory Authentication" on page 11-7

Storing Passwords for Authenticating to Oracle Components

Oracle components store both passwords and password verifiers in Oracle Internet Directory. This section contains these topics:

- About Password Verifiers
- Attributes for Storing Password Verifiers
- Example: How Password Verification Works
- Managing Password Verifier Profiles by Using Oracle Directory Manager
- Managing Password Verifier Profiles by Using Command-Line Tools

About Password Verifiers

Oracle components can store their password values in Oracle Internet Directory as password verifiers. A password verifier is a hashed version of a clear text password. This hashed version is then encoded as a BASE64 encoded string.

You can choose one of these hashing algorithms to derive a password verifier:

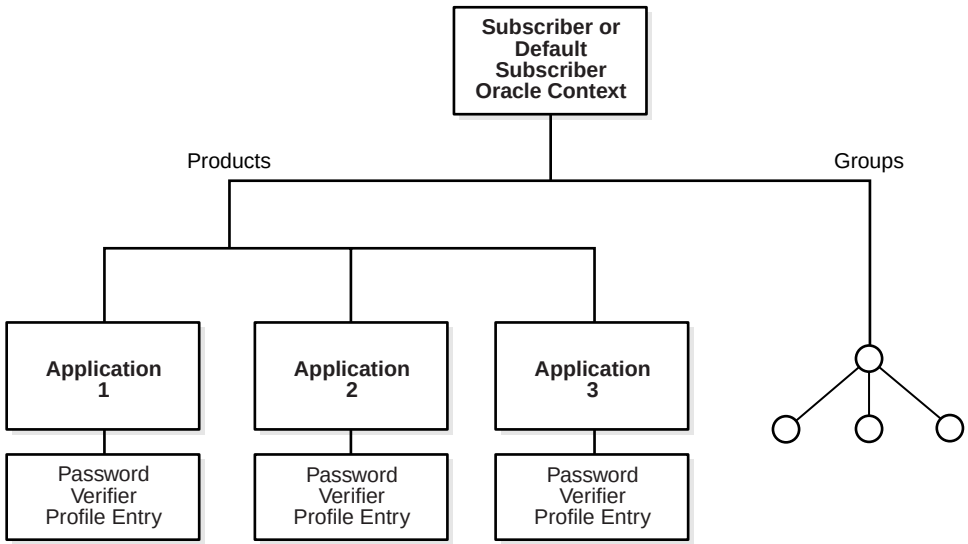
- **MD4**—An improved, and more complex, version of MD4
- **SHA**—Secure Hash Algorithm, which produces a 160-bit hash, longer than MD5. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.
- **UNIX Crypt**—The UNIX hashing algorithm
- SASL/MD5—Simple Authentication and Security Layer/MD5, which adds authentication support to connection-based protocols and uses a challenge-response protocol.
- O3LOGON—A proprietary Oracle algorithm for generating verifiers. It is similar to SASL/MD5 in that it uses a challenge-response protocol.
- ORCLWEBDAV—A proprietary algorithm identical to SASL/MD5 which takes the username in the format username@subscriber.

- ORCLLM—Oracle's representation of the SMBLM algorithm. The SMBLM algorithm is Oracle's representation of the LM variant of the SMB / CIFS challenge / response authentication algorithm.
- ORCLNT—Oracle's representation of the SMBNT algorithm. The SMBNT algorithm is Oracle's representation of the NT variant of the SMB / CIFS challenge / response authentication algorithm.

During Oracle application installation, the Oracle Universal Installer creates for that application a password verifier profile entry containing all the necessary password verification information. It places this entry as shown in Figure 17–1: immediately below the application entry, which resides under the products entry, which, in turn, resides under the subscriber-specific or default Oracle context.

This verifier profile entry is applicable only to users under the given subscriber. It does not apply to users under a different subscriber. The `orclcommonusersearchbase` attribute in the common entry of the subscriber Oracle context must be set to the appropriate value for the verifier generation to be successful. This attribute must be set before verifier generation can take effect.

Figure 17–1 Location of the Password Verifier Profile Entry



Attributes for Storing Password Verifiers

Unlike the directory, which stores user passwords in the `userPassword` attribute, Oracle components store user password verifiers in one of two password attribute types—`authPassword` and `orclPasswordVerifier`—within the user entry. Each attribute type has `appID` as an attribute subtype. The `appID` attribute is a unique identifier representing an Oracle application server or authenticating identity. It is generated during application installation. For example, the appID can be the ORCLGUID of the application entry. This uniquely identifies a particular application.

Table 17–1 Attributes for Storing Password Verifiers in User Entries

Attribute	Description
`authPassword;appID`	A password for authenticating a user to an application. The password value is the same as that used for authenticating the user to the directory, and is synchronized with it. For example, `userpassword`. Several different applications can require the user to enter the same clear text password used for the directory, but each application may hash it with a different algorithm. In this case, the same clear text password can become the source of several different password verifiers. This attribute is multivalued and can contain all the other verifiers that different applications use for this user's clear text password. If the `userpassword` is modified, the `authpasswords` for all applications are regenerated.
`orclPasswordVerifier; appID`	A password for authenticating a user to an application. However, unlike passwords stored in the `authPassword` attribute, it is different from that for authenticating to the directory, and is not synchronized with it. Like `authPassword`, this attribute is multivalued and can contain all the other verifiers that different applications use for this user's clear text password.

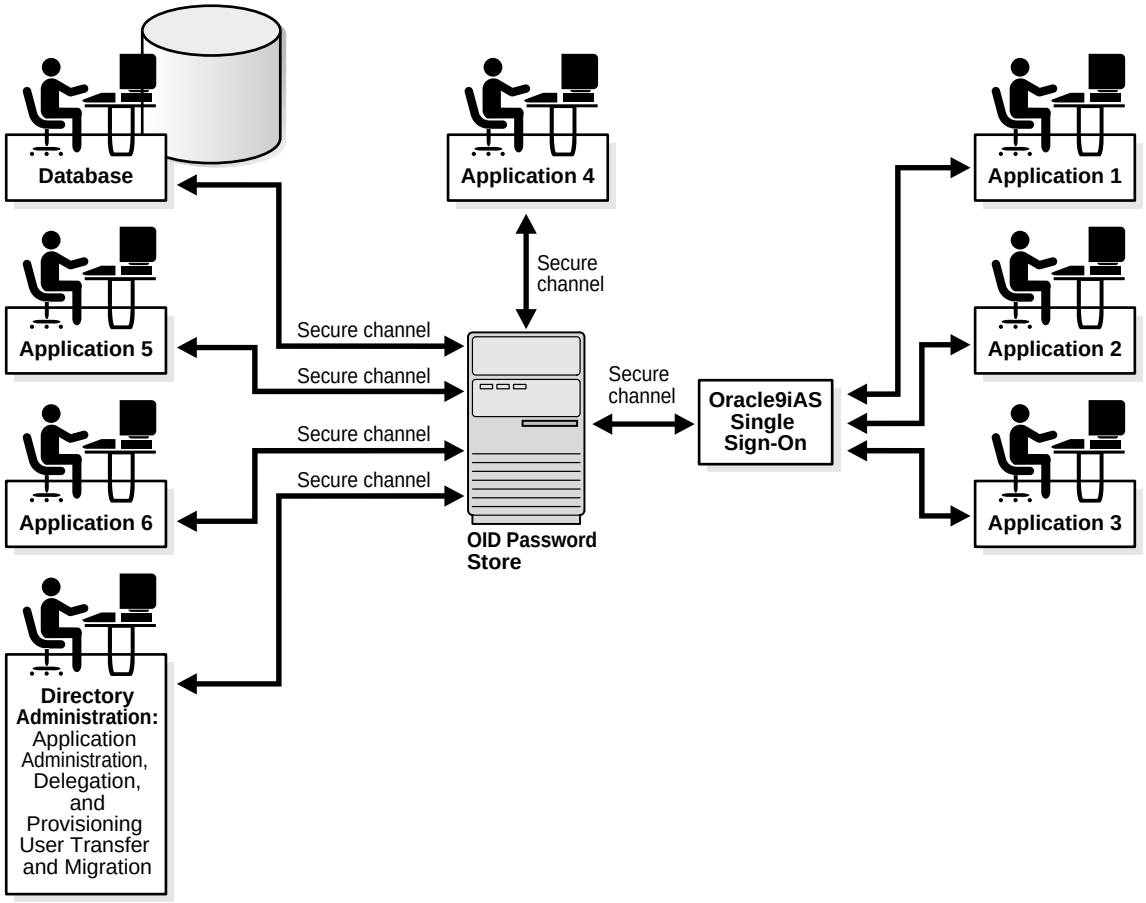
In Figure 17–2, various Oracle components store their password verifiers in Oracle Internet Directory. Oracle9*i*AS Single Sign-On uses the same password as that for the directory, and hence stores it in the `userPassword` attribute.The other applications use different passwords and hence store their verifiers in `orclPasswordVerifier` attribute.

The following is an example of an Application-Verifier Profile:

```
dn:
```

```
cn=IFSVerifierProfileEntry,cn=IFS,cn=Products,cn=OracleContext,o=Oracle,dc=com
objectclass:top
objectclass:orclpwdverifierprofile
cn:IFSVerifierProfileEntry
orclappid:8FF2DFD8203519C0E034080020C34C50
orclpwdverifierparams;authpassword: crypto:SASL/MDS $ realm:dc=com
orclpwdverifierparams;orclpasswordverifier: crypto:ORCLLM
```

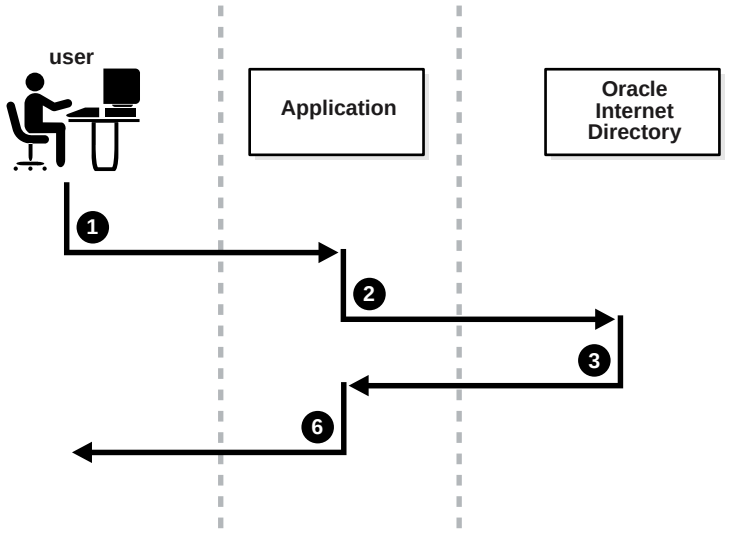
Figure 17–2 Authentication Model



Example: How Password Verification Works

Figure 17–3 shows an example of password verification. In this example, the Oracle component stores its password verifiers in the directory.

Figure 17–3 How Password Verification Works



1. The user tries to log in to an application by entering a user name and a clear text password.
2. The application sends the clear text password to the directory server. If the application stores password verifiers in the directory, then the application requests the directory server to compare this password value with the corresponding one in the directory.
3. The directory server generates a password verifier by using the hashing algorithm specified for the particular application. It compares this password verifier with the corresponding password verifiers the directory. It then notifies the application of the results of the compare operation. For the compare operation to be successful, the application must provide its appID as the subtype of the verifier attribute. For example:

   ```
   ldapcompare -p389 -D "<dn of the app entity>" -w "<password>" -b "<dn of the user>" -a orclpasswordverifier; <appID> -v <password of the user>
   ```

4. Depending on the message from the directory server, the application either authenticates the user or not.

If an application does not use the compare operation, then it simply retrieves from the directory the hashed value of clear text password as entered by the user. The application then compares that value with the hash value it computes. If the two values match, then the application authenticates the user.

Managing Password Verifier Profiles by Using Oracle Directory Manager

You can use Oracle Directory Manager to view and modify password verifier profile entries.

Viewing and Modifying a Password Verifier Profile by Using Oracle Directory Manager

To view an application's password verifiers:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*, then select Password Verifier Management. The right pane displays two columns:
   - Path to Password Verifier Entry column lists the full DN of each password verifier profile entry
   - Password Verifier Entry column lists the corresponding RDNs of each password verifier profile entry
2. Choose the password verifier you want to view. This displays the Password Verifier Profile dialog box for that password verifier. Table 17–2 lists and describes the fields in this dialog box.
3. To modify the hashing algorithm used to generate a password verifier, enter the new value in the OrclPwdVerifierParams field as described in Table 17–2.

Table 17–2 Password Verifier Profile Dialog Box

Field	Description
Path to Password Verifier Entry	The full DN of this password verifier entry. Use this to locate a particular password verifier entry. You cannot modify this field.
Password Verifier Entry	RDN of this password verifier. You cannot modify this field.
Application ID	The unique identifier of the Oracle application. It is generated during application installation. You cannot modify this field.

Table 17–2 Password Verifier Profile Dialog Box

Field	Description

Oracle Password Parameters	Parameters containing information for generating this password verifier. Use this field to specify the hashing algorithm for this password verifier. The syntax is:
	`crypto:`*`hashing_algorithm`*
	For example, if you are using the ORCLLM hashing algorithm, then you would enter:
	`crypto:ORCLLM`
	If you are using SASL/MD5, for example, you can enter the following:
	`crypto:SASL/MD5 $ realm:dc=com`

Managing Password Verifier Profiles by Using Command-Line Tools

Viewing a Password Verifier Profile by Using Command-Line Tools

To view an application's password verifier, perform a search specifying the DN of the password verifier profile.

Modifying a Password Verifier Profile by Using Command-Line Tools

The following example changes the hashing algorithm in an application password verifier profile entry. This password verifier synchronizes with the user's directory password.

```
ldapmodify -p 389 -h my_host -v <<EOF
dn: cn=MyAppVerifierProfileEntry,cn=MyApp,cn=Products,cn=OracleContext,
    o=my_company,dc=com
changetype: modify
replace: orclPwdVerifierParams
orclPwdVerifierParams;authPassword: crypto:SASL/MD5 $ realm:dc=com
EOF
```

18

Password Policies

This chapter discusses password policies—that is, sets of rules that govern how passwords are used.

This chapter contains these topics:

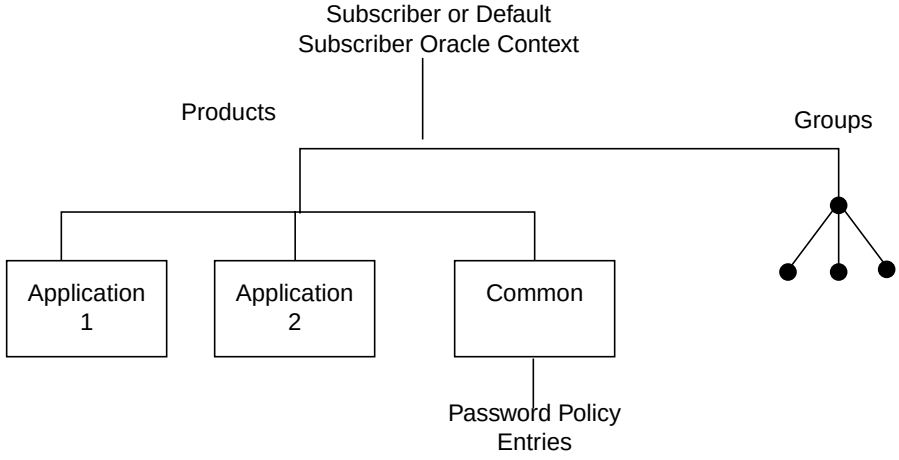
About Password Policies

Password polices are sets of rules that govern how passwords are used. The directory server enforces the password policy syntax checks during `ldapadd` and `ldapmodify` to ensure that the user password meets the requirements set in that policy. The password policy state checks are enforced by the directory server during `ldapbind` and `ldapcompare`. When you establish a password policy, you set the following types of rules, to mention just a few:

- The maximum length of time a given password is valid
- The minimum number of characters a password must contain
- The number of numeric characters required in a password

During Oracle Internet Directory installation, the Oracle Universal Installer creates for each subscriber a password policy entry containing all the necessary password policy information. It places this entry as shown in Figure 18–1: immediately below the common entry, which resides under the products entry, which, in turn, resides under the subscriber or default subscriber Oracle context. This password policy is applicable to all users under a given subscriber. The Oracle Internet Directory password policy is applicable only to the `userpassword` attribute. The orclcommonusersearchbase attribute in the common entry of the subscriber Oracle context must be set to the appropriate value for the password policy to be enforced. This attribute must be set before any password policy modification can take effect.

Figure 18–1 Location of Password Policy Entries



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

Policy	Attribute	Description
Password Expiry Time	`pwdMaxAge`	The maximum length of time, in seconds, that a given password is valid. If this attribute is not present, or if the value is `0` (zero), then the password does not expire. By default, the passwords expire in 60 days.
Password Expiration Warning	`pwdExpireWarning`	The number of seconds before password expiration that the directory server sends the user a warning. If password expiration is enabled, then, by default, the directory server sends no warnings before the password expires. The directory server sends the warning at each logon. If the user does not modify the password before it expires, the user is locked out until the password is changed by the administrator. For this feature to work, the client application must support it. The default is 0, which means no warnings are sent.
Grace Login Limit	`pwdGraceLoginLimit`	Maximum number of grace logins allowed after a password expires. By default, no grace logins.are allowed. The default value is 0.

Policy	Attribute	Description
Password Lockout	pwdLockout	Specification for whether users are locked out of the directory after the number of consecutive failed bind attempts specified by pwdmaxFailure. If the value of this policy attribute is 1, then users are locked out. If this attribute is not present, or if the value is 0, then users are not locked out and the value of pwdMaxFailure is ignored. By default, account lockout is enforced. The account is locked after three consecutive login failures.
Password Maximum Failure	pwdMaxFailure	The number of consecutive failed bind attempts after which a user account is locked. If this attribute is not present, or if the value is 0 (zero), then the account is not locked due to failed bind attempts, and the value of the password lockout policy is ignored. The default is 4.
Password Failure Count Interval	pwdFailureCountInterval	The number of seconds after which the password failure times are purged from the user entry. If this attribute is not present, or if it has a value of 0, then failure times are never purged. The default is 0.
Lockout Duration	pwdLockoutDuration	The number of seconds a user is locked out of the directory if *both* of the following are true: ■ Account lockout is enabled ■ The user has been unable to bind successfully to the directory for at least the number of times specified by pwdMaxFailure You can set user lockout for a specific duration, or until the administrator resets the user's password. A default value of 0 (zero) means that the user is locked out forever.
Check Password Syntax	pwdCheckSyntax	Specification for whether syntax checking is enforced. If 1, then syntax checking is enforced. The default is enabled.
Minimum Number of Characters of Password	pwdMinLength	The minimum number of characters required in a password. By default, the minimum length is 5; however, the value for this attribute must be at least 1.
Number of Numeric Characters in Password	orclpwdAlphaNumeric	Number of numeric characters required in a password. By default, one numeric character is required. That is, the default value is 1.

Policy	Attribute	Description
Old Password Can Be New Password	orclpwdToggle	Specification for whether a user's old password can become the new one. By default, it can. The default value is 1.
Illegal Values	orclpwdIllegalValues	Multivalued attribute containing the common words and attribute types whose values cannot be used as a valid password. By default, all words are acceptable password values.

Note: All user passwords are assumed to be single-valued, as mentioned in the July 2001 version of the IETF draft: http://ietf.org/internet-drafts/draft-behera-ldap-password-policy-05.txt

To establish a password policy, you use the pwdPolicy auxiliary object class, which contains password policy information for the entire directory. You set these values during installation. An entry of this object class is created during installation. It has this DN: cn=pwdpolicyentry,cn=my_application,cn=products,cn=Oracle Context,o=*my_company*,dc=com. In Release 9.0.2, the policy specified applies to the DIT of a given subscriber. Each subscriber can have their own password policy.

This object class contains the following attributes.

Table 18–1 pwdPolicy Object Class Attributes

- pwdMaxAge
- pwdGraceLoginLimit
- orclpwdAlphaNumeric
- pwdLockout
- pwdMinLength
- orclpwdToggle
- pwdLockoutDuration
- pwdCheckSyntax
- orclpwdIllegalValues

- pwdMaxFailure
- pwdFailureCountInterval
- pwdExpireWarning

The default value for each of these attributes is 0 (zero). These attributes are single-valued, except orclpwdIllegalValues, which is multi-valued.

In addition, the object class top contains these operational attributes, to maintain the user-password state information for each user entry.

- pwdChangedtime: The timestamp of the user password creation or modification
- pwdExpirationWarned: The time at which the first password expiration warning is been sent to the user
- pwdFailuretime: The timestamp of consecutive failed login attempts by the user
- pwdAccountLockedTime: The time at which the user account was locked
- pwdReset: Requirement for the user to change the password, if this attribute is enabled
- pwdGraceUseTime: The time stamps of each grace login by the user

> **See Also:** The July 2001 version of the following IETF draft: http://ietf.org/internet-drafts/draft-behera-ldap-password-policy-05.txt

Managing Password Policies by Using Oracle Directory Manager

During Oracle Internet Directory installation, a password policy entry is created for each subscriber. Table 18–2 lists and describes the password policy fields in Oracle Directory Manager.

Table 18–2 Password Policy Fields in Oracle Directory Manager

Field	Description
Password Policy Entry	This field displays the RDN of the password policy entry. You cannot edit this field.

Table 18–2 Password Policy Fields in Oracle Directory Manager

Field	Description
Password Expiry Time	Enter the number of seconds that a given password is valid. If this attribute is not present, or if the value is 0, then the password does not expire. By default, user passwords never expire.
Account Lockout	From the list, select Enable or Disable.
Account Lockout Duration	Enter the number of seconds a user is locked out of the directory if both of the following are true: ■ Account lockout is enabled ■ The user has been unable to bind successfully to the directory for at least the number of times specified by `pwdMaxFailure` You can set user lockout for a specific duration, or until the administrator resets the user's password. A default value of `0` (zero) means that the user is locked out forever.
Password Maximum Failure	Enter the number of consecutive failed bind attempts after which a user account is locked.
Password Failure Count Interval	Enter the number of seconds after which the password failure times are purged from the user entry.
Password Expiration Warning	Enter the length of time before password expiration that the directory server sends the user a warning. By default, no warnings are sent. The directory server sends the warning at each logon. If the user does not modify the password before it expires, then the directory server enforces the modification. This means that the user is locked out until the password is changed by the administrator. For this feature to work, the client application must support it.
Check Password Syntax	Specify whether syntax checking is enforced. If 1, then syntax checking is enforced.
Need to Supply Old Password When Modifying Password	Specify whether user must supply old password with new one when modifying password. By default, the old password is not required.
Minimum Number of Characters of Password	Specify the minimum number of characters required in a password.
Number of Numeric Characters in Password	Specify the number of numeric characters required in a password.

Table 18–2 Password Policy Fields in Oracle Directory Manager

Field	Description
Old Password Can Be New Password	Specify whether a user's old password can become the new one. If you choose Enable from the list, then the old password can become the new one.

When you create a subscriber, you also configure that subscriber's password policies. Later, you can use Oracle Directory Manager to view, refresh, and modify those policies. However, you cannot add or delete them.

Viewing a Subscriber's Password Policies by Using Oracle Directory Manager

To view a subscriber's password policies, in the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Password Policy Management. The navigator pane displays the subscriber password policy entries. The right pane displays a table with two columns:

- The Path to Password Policy Entry column lists the full DN of each password policy entry
- The Password Policy Entry column lists the corresponding RDNs of those policies

For the latest updates to a subscriber's password policies, choose Refresh.

For a particular subscriber's password polices, in the navigator pane, choose the subscriber password policy you want to view.

Modifying a Subscriber's Password Policies by Using Oracle Directory Manager

To modify a subscriber's password policies:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Password Policy Management.
2. In the navigator pane, choose the subscriber password policy you want to modify.
3. In the right pane, modify the attribute fields for that policy.
4. When you are finished, choose Apply.

Managing Password Policies by Using Command-Line Tools

This section contains these topics:

Setting Password Policies by Using Command-Line Tools

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

The file `my_file.ldif` contains:

```
dn: cn=pwdpolicyentry,cn=common,cn=products,cn=OracleContext,o=my_company,dc=com
changetype:modify
replace: pwdlockout
pwdlockout: 1
```

The following command loads this file into the directory:

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

Managing a Subscriber's Password Policies Using Command-Line Tools

Examine the following examples to learn how to view and modify a subscriber's password policies by using command-line tools.

Example: Viewing a Subscriber's Password Policies Using Command-Line Tools

The following example retrieves a specific password policy entry.

```
ldapsearch -p 389 -h my_host -b
"cn=pwdpolicyentry,cn=common,cn=products,cn=OracleContext,o=my_company,dc=com"
-s base "objectclass=*"
```

The following example retrieves all password policy entries:

```
ldapsearch -p 389 -h my_host -b "" -s sub "objectclass=pwdpolicy"
```

Example: Modifying a Subscriber's Password Policies Using Command-Line Tools

The following example modifies a password policy entry.

```
ldapmodify -p 389 -h my_host -v <<EOF
```

```
dn: cn=pwdpolicyentry,cn=common,cn=products,cn=OracleContext,o=my_company,dc=com
changetype: modify
replace: pwdMaxAge
pwdMaxAge: 100000
```

Error Messages

See: "Password Policy Violation Error Messages" on page H-9

19

Capacity Planning Considerations

Capacity planning is the process of assessing applications' directory access requirements and ensuring that the Oracle Internet Directory has adequate computer resources to service requests at an acceptable rate. This chapter explains what you need to consider when doing capacity planning. It guides you through an example of a directory deployment for an email messaging application in a hypothetical company called Acme Corporation

This chapter contains these topics:

About Capacity Planning

If Oracle Internet Directory and the corresponding Oracle9*i* database are running on the same computer, then these are the configurable resources that capacity planners need to consider:

- I/O subsystem (the type and size)
- Memory
- Network connectivity
- CPUs (speed and quantity)

When you plan to acquire hardware for Oracle Internet Directory, you should ensure that all components—such as CPU, memory, and I/O—are effectively used. Generally, good memory usage and a robust I/O subsystem are sufficient to keep the CPU busy.

Any new installation of the Oracle Internet Directory needs two things to be successful:

- Adequate hardware resources so that the installed system can satisfy user demands at peak load rates
- A well tuned system—hardware and software—that makes the best use of available resources, one that squeezes the maximum performance out of available hardware

We begin by looking at an example of a directory deployment for an email messaging application in a hypothetical company called Acme Corporation. As we examine each component of the capacity plan, we will apply our recommendations to the example of Acme Corporation.

The following terms are used throughout this chapter:

- Throughput

  The overall rate at which directory operations are being completed by Oracle Internet Directory. This is typically represented as "operations per second."

- Latency

  The time a client has to wait for a given directory operation to complete

- Concurrent clients

  The total number of clients that have established a session with Oracle Internet Directory

- Concurrent operations

  The amount of concurrent operations that are being executed on the directory from all of the concurrent clients. Note that this is not necessarily the same as the concurrent clients because some of the clients may be keeping their sessions idle.

Getting to Know Directory Usage Patterns: A Case Study

The ability to assess the potential load on Oracle Internet Directory is very important for developing an accurate capacity plan. Let us examine the email messaging software employed by our hypothetical company, Acme Corporation. The email messaging software in this example is based on Internet Message Access Protocol (IMAP). There are two main types of software that access Oracle Internet Directory:

- The IMAP clients, which will validate email addresses within the company before sending the mail to the IMAP server. These clients include software programs like Netscape Messenger and Microsoft Outlook.
- The messaging software itself, also called the Mail Transfer Agent (MTA), which will look up the directory to route mail from the outside world to internal mailboxes as well as route internal mails to company-wide distribution lists.

Let us assume that the private aliases and private distribution lists of individual users are also stored in the directory. Let us further make the following assumptions, which will allow us to guess the size of the directory:

Entry Type	Size
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:

Entry Type	Size
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.

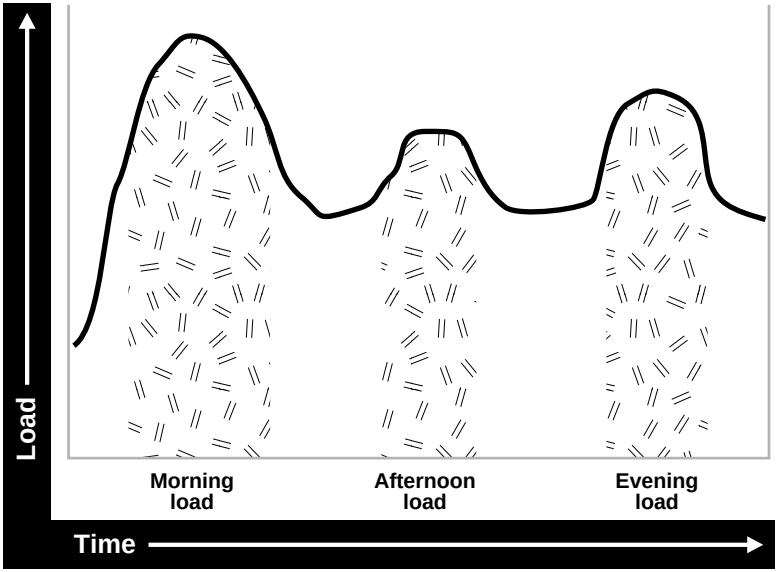
The following table summarizes all the possible directory lookups that can happen in one day:

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 19–1.

Figure 19–1 Usage Analysis of Current Email System



The email system and Oracle Internet Directory are maximally stressed in the mornings. There are other usage peaks as well—one close to lunch time, and one near the end of business day. However, it is in the mornings that the Oracle Internet Directory is stressed the most.

Let us assume that 90 percent of all the directory lookups happen during normal working hours. Let us now split up the working hour load into the following categories (assuming an 8 hour workday):

Shift Load	Lookups
Morning load	65%: 0.90 x 0.65 x 3,200,000 = 1,872,000 lookups for 2 hours (936,000 lookups per hour)
Afternoon load	10%: 0.90 x 0.10 x 3,200,000 = 288,000 lookups for 1 hour (288,000 lookups per hour)
Evening load	20%: 0.90 x 0.20 x 3,200,000 = 576,000 lookups for 2 hours (288,000 lookups per hour)

The above calculations indicate that the Oracle Internet Directory in this case should be designed to handle the peak load of 936,000 lookups per hour.

Now that we know the data-set size as well as the performance requirements, we can now look into individual components of the installation and estimate good values for each.

I/O Subsystem Requirements

This section contains these topics:

About the I/O Subsystem

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 20, "Tuning Considerations". Depending on the tolerance of disk failures, different levels of Redundant Arrays of Inexpensive Disks (RAID) can also be considered.

Assuming that the decision has been made to get the best possible I/O subsystem, we focus the next section on deriving sizing estimates for the disks themselves.

Rough Estimates of Disk Space Requirements

You can use the following table to derive a rough estimate of the overall disk requirement:

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

The data shown in the previous table makes the following assumptions:

- There are about 20 cataloged attributes.
- There are about 25 attributes per entry.
- The average size of an attribute is about 30 bytes.

Going back to our example of Acme Corporation, since our directory population is about one million, this would imply that our disk requirements are approximately 4.5 GB to 6.5 GB. Note that the assumptions made for Acme Corporation regarding

the number of cataloged attributes are different, but the previous table should give an approximate figure of the size requirements.

Since the directory may be deployed for a wide variety of applications, these assumptions need not necessarily hold true for all possible situations: There might be cases where the size of attributes is large, the number of attributes per entry is large, extensive use of ACIs has been made, or the number of cataloged attributes is very high. For such cases, we present simple arithmetic procedures in the following section which will allow the planners to get a more detailed perspective of their disk requirements.

Detailed Calculations of Disk Space Requirements

Because Oracle Internet Directory stores all of its data in an Oracle9*i* database, the sizing for disk space is primarily a sizing of the underlying database. Oracle Internet Directory stores its data in the following tablespaces:

Tablespace Name	Contents
OLTS_ATTR_STORE	Stores all of the attributes for all entries in the DIT
OLTS_IND_ATTRSTORE	Stores the indices pertaining to attributes in the directory
OLTS_CT_DN	Stores the distinguished name catalog
OLTS_IND_CT_DN	Stores the indices pertaining to the DN catalog
OLTS_CT_CN	Stores the common name catalog
OLTS_CT_OBJCL	Stores the ObjectClass catalog
OLTS_CT_STORE	Stores all the remaining (including user-defined) catalogs
OLTS_IND_CT_STORE	Stores the indices pertaining to the user-defined catalogs
P1TS_ATTRSTORE	Stores the catalogs and the attributes table for server manageability
P1TS_IND_ATTRSTORE	Stores the indices on the tables acquired by OID Server Manageability
OLTS_DEFAULT	Stores all of the data pertaining to the administration of the Oracle Internet Directory as well as the data used for replication support
OLTS_TEMP	Used for creating various indices on the tables. It should be large enough so that all index creations can go through.
SYSTEM	Required by Oracle9*i* database for various book-keeping purposes. Typically, its size remains constant at about 300MB.

This section presents simple arithmetic procedures to determine the size requirements of each of the tablespaces shown above. All of the size calculations are based on the following variables:

Variable Name	Description
num_entries	Total number of entries in the directory
attrs_per_entry	Average number of attributes per directory entry
avg_attr_size	Average size of the attribute value in bytes
avg_dn_size	Average size of the DN of an attribute in bytes
objectclass_per_entry	Average number of object classes that an entry belongs to
objectclass_size	Average size of the name of each objectclass in bytes
num_cataloged_attrs	Number of cataloged attributes used in the entries
entries_per_catalog	Average number of entries per catalog table. This is required because not all cataloged attributes will be present in all entries in the DIT.
change_log_capacity	Number of changes that we wish to buffer for replication purposes
num_acis	Overall number of ACIs in the directory
num_auditlog_entries	Number of auditlog entries to store in the directory
db_storage_ovhd	Overhead of storing data in tables. This overhead corresponds to the relational constructs as well as operating system specific overhead. A value of 1.3 for this variable would represent a 30 percent overhead. The minimum value for this variable is 1.
db_index_ovhd	Overhead of storing data in indices. This overhead corresponds to the relational constructs as well as the operating system specific overhead. A value of 5 for this variable would represent a 400 percent overhead. The minimum value of this variable is 1.
factor_of_safety	Multiplier for accommodating growth and errors in calculations. A value of 1.3 for this variable would represent a 30 percent factor of safety. The minimum value for this variable is 1.
initial_num_entries	Total number of entries that are initially bulk-loaded into the directory
avg_attrname_len	Average size of attribute name, in bytes

Variable Name	Description
num_stats_entries	Number of statistics entries generated by OID Server Manageability when the host DSF attribute 'orclstatsflag' is enables
attrs_per_stats_entry	Average number of attributes per statistics entry

Using the variables shown in the preceding table, the size of individual tablespaces can be calculated as follows:

Tablespace Name	Size
OLTS_ATTR_STORE	num_entries * {((attrs_per_entry) * (avg_attrnam_len + avg_attr_size + 22)) + 6*35 } * db_storage_ovhd
OLTS_IND_ATTRSTORE	num_entries * (attrs_per_entry + 6) * 20
OLTS_CT_DN	num_entries * 2 * (avg_dn_size + 4)
OLTS_IND_CT_DN	num_entries * 2 * (avg_dn_size * 3)
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)
P1TS_ATTRSTORE	num_stats_entries * ((avg_attrnam_len + avg_ attr_size + 20) * attrs_per_stats_entry) * db_ storage_ovhd * (orclstatsperiodicity/60) * 12)
P1TS_IND_ATTRSTORE	(num_stats_entries * attrs_per_stats_entry * 20) * ((orclstatsperiodicity/60) * 12)

Tablespace Name	Size
OLTS_DEFAULT	`(change_log_capacity * 4 * avg_attr_size * db_storage_ovhd * db_index_ovhd) + initial_num_entries * 2 * (avg_dn_size + 4)`
OLTS_TEMP	`(size of OLTS_IND_ATTR_STORE) + (size of OLTS_IND_CT_STORE)`
SYSTEM	`300 MB`

Using the arithmetic operations shown in the preceding table, one can compute the exact space requirements for a wide variety of Oracle Internet Directory deployment scenarios. The sum of the sizes of each of the tablespaces should yield the overall database disk requirement. One can optionally multiply that by the "factor_of_safety" variable to get a figure that can compensate for unforeseen circumstances.

Going back to our example of Acme Corporation, we can assign values to each of the variables based on the requirements stated in previous sections. The following table illustrates the values of each variable introduced in this section for Acme Corporation.

Variable Name	Value
num_entries	1,000,000
attrs_per_entry	20
avg_attr_size	32 bytes
avg_dn_size	40 bytes
objectclass_per_entry	5 (each entry belongs to an average of 5 object classes)
objectclass_size	10 bytes
num_cataloged_attrs	10
entries_per_catalog	1,000,000
change_log_capacity	80,000 changes (2 per user)
num_acis	80,000 ACIs (2 per user)
num_auditlog_entries	1000
db_storage_ovhd	1.4 (40% overhead)
db_index_ovhd	5.0 (400% overhead)

Variable Name	Value
factor_of_safety	1.5 (50% factor of safety)
initial_num_entries	1,000,000
num_stats_entries	5
attrs_per_stats_entry	12
'orclstatsperiodicity'	60 (root DSE attribute)
avg_attrname_len	6

If we now plug these values into the equations described earlier, we get the following values:

Tablespaces Name	Size in Bytes	Size in MB
OLTS_ATTRSTORE	2076180480	1980
OLTS_IND_ATTRSTORE	545259520	520
OLTS_CT_DN	92274688	88
OLTS_IND_CT_DN	251658240	240
OLTS_CT_CN	57671680	55
OLTS_CT_OBJCL	71303168	68
OLTS_CT_STORE	530579456	506
OLTS_IND_CT_STORE	1918894080	1830
P1TS_ATTRSTORE	104857600	100
P1TS_IND_ATTRSTORE	52428800	50
OLTS_DEFAULT	170917888	163
OLTS_TEMP	2533359616	2416
SYSTEM	314572800	300
Total Size	**8719958016**	**8316**

The table above shows that the estimated size of the database for Acme Corporation would be about 8.25 GB. If all of the data is being loaded in bulk, then the bulkload tool of Oracle Internet Directory would require an additional 30 percent of space

occupied by the database to store its temporary files. For Acme Corporation, this would add about 2.5 GB to the total space requirement.

Memory Requirements

Memory is used for a number of distinct tasks by any database application, including Oracle Internet Directory. If memory resources are insufficient for any of these tasks, the bottleneck causes the CPUs to work at lower efficiency and system performance to drop. Furthermore, memory usage increases in proportion to the number of concurrent connections to the database and the number of concurrent users of the directory.

The memory available to processes comes from the virtual memory on the system, which is somewhat more than available physical memory. If the sum of all active memory usage exceeds the available physical memory on the system, the operating system may need to store some of the memory pages on disk. This is called paging. Paging can degrade performance if memory is too oversubscribed. Generally, you should not exceed 20 percent over-subscription of physical memory. If paging occurs, you need either to scale back memory usage by processes or to add more physical memory. Keep in mind the trade-offs: There are physical limits to the amount of memory you can add, but scaling back on per-process memory usage can significantly degrade performance.

The main consumers of memory are the database buffer cache within the system global area (SGA) and the OID Server Entry Cache (if enabled). Getting a good hit ratio for the buffer cache and the entry cache requires allocating enough memory in each area. The following formula gives a rough estimate for the amount of RAM required to cache 'N' entries in the entry cache:

’N’ * (attrs_per_entry + 6) * (avg_attrname_len + avg_attr_size + 72)

> **See Also:** Chapter 20, "Tuning Considerations" for further information on SGA tuning

The following table gives minimum memory requirements for different directory configurations:

Directory Type	Entry Count	Minimum Memory
Small	Less than 600,000	512 MB
Medium	600,000 to 2,000,000	1 GB

Directory Type	Entry Count	Minimum Memory
Large	Greater than 2,000,000	2 GB

Going back to our example of Acme Corporation, the number of entries in the directory are close to 1,000,000 (1 million). Oracle Corporation recommends choosing the 2 GB option in order to maximize performance.

Network Requirements

The network is rarely a bottleneck in most installations. However serious consideration must be given to it during the capacity planning stage. If the clients do not get adequate network bandwidth to send and receive messages from Oracle Internet Directory, the overall throughput will seem to be very low. For example, if we have configured Oracle Internet Directory to service 800 search operations per second, but the computer running the Oracle directory server is only accessible through a 10 Mbps network (10-Base-T switched ethernet), and we have only 60 percent of the bandwidth available, then the clients will only see a throughput of 600 search operations a second (assuming each search operation causes 1024 bytes to be transferred on the network). The following table shows the maximum possible throughput (in operations per second) for two types of operations (one requiring a transfer of 1024 bytes the other requiring a transfer of 2048 bytes) for two types of networks, 10 Mbps & 100 Mbps, at different rates of bandwidth availability:

Percent Available Bandwidth	Operations/sec 1024 bytes		Operations/sec 2048 bytes	
	10 Mbps	100 Mbps	10 Mbps	100 Mbps
30	300	3000	150	1500
40	400	4000	200	2000
50	500	5000	250	2500
60	600	6000	300	3000
70	700	7000	350	3500
80	800	8000	400	4000
90	900	9000	450	4500

In some cases, it may also be important to consider the network latency of sending a message from a client to the Oracle directory server. In some WAN implementations, the network latencies may become as high as 500 milliseconds, which may cause the clients to time out for certain operations. In summary, given a range of networking options, the preferred choice should always be for highest bandwidth, lowest latency network.

Going back to the example of Acme Corporation, their peak usage rate is 936,000 lookups per hour which results in an equivalent number of lookup operations to the directory. This requires about 260 directory operations per second. Assuming that each operation results in a transfer of 2 KB of data on the network, this would imply that we should have a 100 Mbps network or at least 60 percent bandwidth available on a 10 Mbps network. Since the 100 Mbps network will typically have a lower latency, we will chose that over the 10 Mbps network.

CPU Requirements

This section contains these topics:

CPU Configuration

The CPU sizing for Oracle Internet Directory is directly a function of the user workload. The following factors will determine CPU configuration:

- The number of concurrent operations you want to support. This will be directly dependent on the number of users performing operations simultaneously.
- The acceptable latency of each operation. For example, in an email application, a latency per operation of 100 milliseconds might be desirable, but in most cases a latency of 500 milliseconds might still be acceptable.

CPU resources can be added to a system as the workload increases, but these additions seldom bring linear scalability to all operations since a lot of operations are not purely CPU bound. We classify the processing power of a computer by a performance characteristic that is commonly available from all vendors, namely, SPECint_rate95 baseline. This number is derived from a set of integer tests and is available from all system vendors as well as the SPEC Web site (http://www.spec.org).

Note: SPECint_rate95 should not be confused with the regular SPECint95 performance number. The SPECint95 performance number gives an idea of the integer processing power of a particular CPU (for systems with multiple CPUs, this number is typically normalized). The SPECint_rate95 gives the integer processing power of an entire system without any normalization.

Because Oracle Internet Directory makes efficient use of multiple CPUs on an SMP computer, we chose to categorize computers based on their SPECint_rate95 numbers. Even within SPECint_rate95 we chose the baseline number as opposed to the commonly advertised result. This is because the commonly advertised result is actually the peak performance of a computer, whereas the baseline number represents the performance in normal circumstances.

Rough Estimates of CPU Requirements

Since Oracle Internet Directory is typically co-resident with the Oracle9*i* database, we recommend at least a two-CPU system. We give the following rough estimates based on the level of usage of Oracle Internet Directory:

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

Detailed Calculations of CPU Requirements

It is difficult to determine the CPU requirements for all operations at a given deployment site since the amount of CPU consumed depends upon several factors, such as:

- The type operation: base search, subtree search, modify, add etc.
- If SSL mode is enabled or not, since SSL consumes an additional 15 to 20 percent of CPU resources.
- If Oracle Internet Directory server entry cache is enabled or not, since the hit ratio affects CPU usage.

- The number of entries returned for a search
- The number of access control policies that need to be checked as part of a search

In most of the cases, except SSL, we can expect that there is a large latency between the Oracle Internet Directory server process and the database. When a thread in the Oracle Internet Directory server process is waiting for the database to respond, other threads within the Oracle Internet Directory server process can be put to work by other client requests needing LDAP server specific processing. As a result, for any mix of operations, one can always come up with a combination of concurrent clients and Oracle Internet Directory server processes that will result in 100 percent CPU utilization. In this case, the CPU becomes the bottleneck.

Given this fact, we have taken a 'messaging' type of subtree search operation and tried to estimate the CPU resources need to support a given number of concurrent operations without degrading the throughput of operations. The 'messaging' search operation involves subtree scope, a simple exact match filter and a result set of one entry. For Oracle Internet Directory Release 9.0.2:

SPECint_rate95 baseline = 0.5 * (max # of concurrent operations at peak throughput)

What this means is that if we need to support 600 concurrent clients without degrading the throughput of operations, then we need a computer that has at least a SPECint_rate95 baseline rating of (0.5 * 600) = 300.

In terms of throughput of operations, for Oracle Internet Directory Release 9.0.2:

SPECint_rate95 baseline = 0.4 * (throughput of operations at max supported concurrency)

What this means is that if we need a throughput of 750 operations per second for the given maximum number of supported concurrent operations, then we need a computer that has at least a SPECint_rate95 baseline rating of (0.4 * 750) = 300.

It has been proven that Oracle Internet Directory scales very well with additional CPU resources. What this means is:

- For a given concurrency of operations, we can achieve higher throughput of operations (and hence, a lower latency) by adding additional CPU resources.

- For a given throughput of operations (and latency), we can support higher concurrency of operations by adding additional CPU resources.

Going back to our example of Acme Corporation, let us assume that we want adequate CPU resources to support 500 concurrent 'messaging' type of subtree search operations with each client seeing subsecond latency. Taking a factor of

safety of 20 percent, our preliminary estimate of CPU requirement would be a computer with a SPECint_rate95 baseline of at least 360.

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: 750 searches / sec at concurrency of 500 clients

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 360.

Several simplifying assumptions were made so that the sizing calculations could be more intuitive.

20

Tuning Considerations

Once you have completed capacity planning as described in Chapter 19, "Capacity Planning Considerations", and you have acquired the necessary hardware, then you must ensure that the combined hardware and software are yielding the desired levels of performance. This chapter gives guidelines for tuning an Oracle Internet Directory installation. It contains these topics:

- About Tuning
- Tools for Performance Tuning
- CPU Usage Tuning
- Memory Tuning
- Disk Tuning
- Database Tuning
- Performance Troubleshooting

About Tuning

The two main performance metrics for any installation of Oracle Internet Directory are:

- The average latency of individual operations at peak load

  This is the time for each operation to complete.

- The overall throughput of Oracle Internet Directory expressed in operations per second at peak load

  This is the rate at which an instance of Oracle Internet Directory is capable of completing client operations

If the performance tests yield poor results, the performance problems may be identified and fixed using the information provided in the following sections.

Tools for Performance Tuning

Knowledge of the following tools is recommended for Solaris and most other UNIX operating systems:

Tool	Description
top	Displays the top CPU consumers on a system
vmstat	Shows running statistics on various parts of the system including the Virtual Memory Manager
mpstat	Shows an output similar to vmstat but split across various CPUs in the system. This is available on Solaris only.
iostat	Shows the disk I/O statistics from various disk controllers

Knowledge of the following tools is recommended for Windows NT:

Tool	Description
Windows NT Performance Monitor	Provides a customized view of the events in the system
Windows NT Task Manager	Provides a high level output (like 'top' on UNIX) of the major things happening in the system.

Knowledge of the following tools is recommended for Oracle9*i*:

- `utlbstat.sql` and `utlestat.sql, or statspack`
- The ANALYZE function in the DBMS_STATS package

See Also:

- *Oracle9i Database Reference* in the Oracle Database Documentation Library for information about `utlbstat.sql` and `utlestat.sql`
- *Oracle9i Database Concepts* in the Oracle Database Documentation Library for information about the ANALYZE function in the DBMS_STATS package

In addition to the operating system tools, the LDAP applications being used in a customer environment must be able to provide latency and throughput measurement.

In addition, the Database Statistics Collection Tool (oidstats.sh), located at `$ORACLE_HOME/ldap/admin`, is provided to analyze the various database 'ods' schema objects to estimate the statistics.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

See Also: "OID Database Statistics Collection Tool Syntax" on page A-55

CPU Usage Tuning

The CPU is perhaps the most important resource available for any software. While Chapter 19, "Capacity Planning Considerations" gives a rough estimate of the required CPU horsepower for a given application load, sometimes insufficient tuning can cause inefficient use of the CPU resources. Consider tuning CPU resources if either of the following cases is true:

- At peak loads the CPU is 100 percent utilized.
- At peak loads the CPU is underutilized, there is a significant amount of idle time in the system, and this idle time cannot be eliminated at even higher loads.

Internal benchmarks show that Oracle Internet Directory performs best when approximately 70 to 75 percent of the CPU resources are consumed by Oracle Internet Directory processes, and the remaining (about 25 to 30 percent) are consumed by the Oracle foreground processes corresponding to the database connections. While monitoring CPU usage, it is also important to monitor the percentage of time spent in the system space compared to user space. Internal benchmarks show best throughput numbers at about 85 percent user and 15 percent system time.

This section contains these topics:

Tuning CPU for Oracle Internet Directory Processes

The demands placed by Oracle Internet Directory processes on the CPU can be controlled by the ORCLSERVERPROCS and ORCLMAXCC parameters. This table lists suggested values for these parameters for various client loads:

ORCLSERVERPROCS	ORCLMAXCC	# Concurrent clients supported without degrading throughput of operations	# Clients supported without dropping connections	Required # of CPUs
1	2	40		1
2	10	400	800	2
4	10	800	1600	4
8	10	1600	3200	8

If we take the example of 500 concurrent clients, a value of 4 for ORCLSERVERPROCS with a value of 10 for ORCLMAXCC will result in the following configuration:

- There will be eleven (10+1) server processes created.
- Each server process will spawn 10 worker threads that will do the actual work.
- Each server process will also maintain a pool of eleven database connections (10+1) that will be shared among the worker threads.

Oracle Internet Directory scales very well with CPU resources both with respect to the throughput of operations and concurrency of clients. From the above table, say we have a 4 CPU box and are able to maintain a peak throughput of 'p' operations per second for a concurrency of 'n' clients.

With additional number of CPUs or with faster CPUs, we can achieve either or both of the following benefits:

- Achieve a thoughput higher than 'p' for the same concurrency of 'n' clients
- Maintain the same 'p' operations throughput for a concurrency higher than 'n'

If the CPU usage at peak loads is not at 100 percent and the system is idle for a large percentage of the time (that is, more than 5 percent), this indicates that Oracle Internet Directory processes are under-configured and are not making the best utilization of the CPU resources. To solve this problem, one must systematically

increase the values of ORCLSERVERPROCS and ORCLMAXCC until the CPU utilization reaches 100 percent and the system and user time are split up as follows:

- User time: 85 percent or higher
- System time: 15 percent or lower

Tuning CPU for Oracle Foreground Processes

Tuning of CPU resources for Oracle Foreground processes should be considered only if both of the following conditions are met:

- The CPU usage is close to 100 percent at peak loads.
- Oracle foreground processes consume more than 30 percent of all available CPU resources.

If Oracle foreground processes are consuming excessive CPU, it implies that the queries that Oracle Internet Directory is making against the database are using too many CPU cycles. Although there is very little control available to the users on the types of underlying operations performed by the database, the following should be attempted:

- Database statistics on all of the tables and indices associated with the ODS user on the database must be collected using the ANALYZE command. This helps the cost-based optimizer make better execution plans for the queries generated by Oracle Internet Directory. `$ORACLE_HOME/ldap/admin/oidstats.sh` can be used to collect statistics.
- 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.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Taking Advantage of Processor Affinity on SMP Systems

Several Symmetric Multi-Processor (SMP) systems offer the capability to bind a particular process to a particular CPU. While it is generally a good idea not to bind any process to any processor, it may improve performance if the following conditions are met:

- The CPU utilization of the entire system is close to 100 percent.
- There are more than two CPUs on the computer.

In internal benchmarks, it has been observed that binding the OID Server process and its associated Oracle shadow processes to the same CPU generally gives the best performance.

Other Alternatives for a CPU Constrained System

If none of the tips stated in the preceding sections solve CPU related performance problems, the following options are available:

- Upgrade the processing power of the computer, that is, add more CPUs or replace slower CPUs with faster ones.
- Keep the Oracle directory server and the associated Oracle9*i* database on separate computers.

Memory Tuning

After the CPU, memory is the next most important thing to tune. The primary consumer of memory in an Oracle Internet Directory installation is the Oracle9*i* database. Make the SGA of the back-end database large enough while leaving room for Oracle Internet Directory and Oracle processes to operate their private stacks and heaps. This section provides some details on determining various components of the SGA.

This section contains these topics:

Tuning the System Global Area (SGA) for Oracle9*i*

The SGA should be sized based on the available physical memory on the system running Oracle9*i*.

> **See Also:** *Oracle9i Database Performance Guide and Reference* in the Oracle Database Documentation Library for more information on determining appropriate sizes for the SGA. This book tells how to ensure that the SGA size does not cause increased paging swapping activity. The latter is very detrimental to performance.

Once the available size of the SGA is determined, two primary tuning items need to be considered:

- Size of the shared pool
- Size of the buffer cache

An initial estimate for the shared pool size is .5 MB per concurrent database connection determined above.

If this estimate consumes more than 30 percent of the total SGA, use 30 percent of the total SGA instead.

Divide 60 percent of the remaining available SGA size by the block size for the database and use this value for the number of DB_BLOCK_BUFFERS. Both of these values should be initial estimates and can be refined using BSTAT/ESTAT and other RDBMS monitoring tools to determine more accurate sizes for best performance.

Other Alternatives for a Memory-Constrained System

If there is insufficient memory to run both the database and the Oracle directory server on the same computer, then one can put the database on a different computer.

Disk Tuning

Balancing Disk I/O is an important consideration in overall RDBMS, and hence Oracle Internet Directory performance. Typically, one can maximize the I/O throughput by using one or more of the following techniques:

- Striping logical volumes so that the I/O operations use multiple disk spindles
- Putting different tablespaces in different logical and physical disk volumes
- Distributing the disk volumes on multiple I/O controllers

> **See Also:** *Oracle9i Database Performance Guide and Reference* in the Oracle Database Documentation Library for general information about balancing and tuning disk I/O

This section contains these topics:

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 20-9 for more discussion about logical disks

Separate the following:

- OLTS_ATTRSTORE and OLTS_IND_ATTRSTORE

  Separating the attribute store table from its index

- OLTS_CT_DN and OLTS_IND_CT_DN

  Separating the DN catalog from its index

- OLTS_xxxx and OLTS_IND_xxxx

  (Empirically, separate the storage tablespace from the associated index)

- OLTS_IND_ATTRSTORE and OLTS_IND_CT_DN

  Alternating the attribute store and DN catalog indexes. This helps even if there are only two logical disks available (one containing OLTS_CT_DN and OLTS_IND_ATTRSTORE and the other containing OLTS_IND_CT_DN and OLTS_ATTRSTORE)

RAID

The information on balancing tablespaces is given in terms of separating Oracle Internet Directory tablespaces onto different logical drives. This assumes that a 'logical drive' is manifested on a separate disk or set of disks from other 'logical drives', and thus represents a division among disks for I/O. (Two logical drives on the same physical disk media do not really provide the same combined I/O throughput of two logical drives located on different physical media.) If a logical

drive can be manifest on a striped or RAID disk subsystem, then this may increase the I/O capacity of that logical drive. However, the tablespace locations considered earlier remain applicable when considering, for instance, different logical drives of a volume manager.

Database Tuning

This section describes the other tunable parameters available to an Oracle Internet Directory installation.

The following table gives a quick overview of the recommended values of RDBMS parameters for various client loads. These parameters are configurable in the initialization parameter file.

Parameters	500 Concurrent LDAP Clients	1000 Concurrent LDAP Clients	1500 Concurrent LDAP Clients	2000 Concurrent LDAP Clients
Open_cursors	200	200	200	200
Sessions	225	600	800	1200
Database_block_ buffers	200 to 250 MB	200 to 250 MB	200 to 250 MB	200 to 250 MB
Database_block_size	8192	8192	8192	8192
Shared_pool_size	30 to 40 MB	30 to 40 MB	30 to 40 MB	30 to 40 MB
Processes	400	800	1000	1500

This section describes each of the RDBMS tunable parameters in more detail. It contains these topics:

- Required Parameter
- Parameters Dependent on Oracle Internet Directory Server Configuration
- SGA Parameters Dependent on Hardware Resources

Required Parameter

Configure the OPEN_CURSORS parameter as follows:

```
OPEN_CURSORS=200
```

The Oracle9*i* default of 50 or so is too small to accommodate Oracle Internet Directory server cursor cache. Note that this value is not dependent on other Oracle

Internet Directory server parameters, such as # SERVERS and # WORKERS. The value of 200 is sufficient for any size DIT.

Parameters Dependent on Oracle Internet Directory Server Configuration

Configure the SESSIONS parameter as follows:

```
PROCESSES = (# OID server processes per instance) x
            (# DB Connections per server + 1) x
            (# of OID instances) + 20
SESSIONS = 1.1 * PROCESSES + 5
```

Each Oracle Internet Directory server process requires a number of concurrent database connections equal to the number of worker threads configured for that server plus one. The total number of concurrent database connections allowed must therefore include this number per server, per instance. The additional 20 connections added to the parameter value accounts for the Oracle background processes plus other Oracle Internet Directory processes such as OID Monitor, OID Control, Oracle directory replication server, and bulk tools.

Using Shared Server Process

Depending on the total number of concurrent database connections required, and as determined by the setting for the SESSIONS parameter, enabling shared server process may help balance overall system load better. If the total number of concurrent database connections required is over 300, then configure the shared server. One shared server should be configured for every 10 database connections required.

Note: The number of required concurrent database connections depends on the hardware selected. See *Oracle9i Net Services Administrator's Guide* and *Oracle9i Database Administrator's Guide*, both in the Oracle Database Documentation Library, for further information about the shared server configuration.

SGA Parameters Dependent on Hardware Resources

The main parameters that contribute to the SGA are discussed in "Memory Tuning" on page 20-7. The following are a few more parameters that may be tuned:

- Sort area

  Set to 262144 (256k) to ensure sufficient sort area available to prevent on-disk sorts.

- Redo Log Buffers

  Set to 32768 (32k) as an initial estimate. If log write performance becomes a performance problem, use a large enough value to make sure (redo log space requests / redo entries) > 1/5000 to prevent the LGWR process from falling behind. This overall has little size effect on the variable SGA size, so making this a little bit too large should not be a problem.

Entry Caching

In the current release of OID, the OID server entry cache is supported only in the single server OID instance. The benefits of entry caching are maximized when the entry cache hit ratio is very high. It is recommended that the entry cache be used for small- to medium-sized directory deployments where two criteria are met:

- the working set of directory entries can be reasonably completely cached, and
- the concurrency of clients is such that it can be handled by a single server OID instance.

Internal benchmarks have indicated that for directory deployments where the working set of entries is a few hundred thousand entries, the entry cache doubled the throughput of operations for up to 1000 concurrent clients.

For larger directory deployments involving a larger working set of directory entries and a higher concurrency of clients, the multi-process OID server instance and the Oracle buffer cache proves to be the scalable architecture of choice for performance.

Performance Troubleshooting

This section gives some quick pointers for common performance related problems.

If LDAP search performance is poor, make sure that:

- The attributes on which the search is being made are indexed

- Schema associated with the ODS user is ANALYZED

  For searches involving multiple filter operands, make sure that the order in which they are given goes from the most specific to the least specific. For example, `&(l=Chicago)(state=Illinois)(c=US)` is better than `&(c=US)(state=Illinois)(l=Chicago)`.

If LDAP add or modify performance is poor, make sure that:

- There are enough redo log files in the database
- The undo tablespace in the database is large enough
- The schema associated with the ODS user is ANALYZED

You can also use the OID Database Statistics Collection tool to analyze the various database ods schema objects to estimate the statistics.

> **See Also:** "OID Database Statistics Collection Tool Syntax" on page A-55 for instructions on using the OID Database Statistics Collection tool

21

High Availability And Failover Considerations

This chapter discusses the high availability and failover features and deployment guidelines for Oracle Internet Directory. It contains these topics:

- About High Availability and Failover for Oracle Internet Directory
- Oracle Internet Directory and Oracle9i Technology Stack
- Failover Options on Clients
- Failover Options in the Public Network Infrastructure
- Availability and Failover Capabilities in Oracle Internet Directory
- Failover Options in the Private Network Infrastructure
- High Availability Deployment Examples

> **See Also:** Part VI, "The Directory and Clusters" for information about high availability and failover in clustered environments

About High Availability and Failover for Oracle Internet Directory

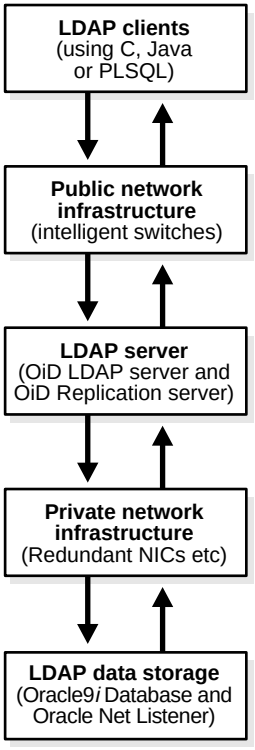
Oracle Internet Directory is designed to address the deployment needs of mission critical applications requiring a high degree of system availability. To achieve a high degree of availability, all components in the system must facilitate redundancy, and all interfaces must facilitate failure recognition and recovery, called **failover**. In addition, integration of application independent network failover capabilities in the overall deployment is also essential to achieve overall system availability.

Oracle products are commonly targeted for high availability environments and hence necessary capabilities are built into all layers of the Oracle technology stack described on page 21-2. Typically, it is not necessary to employ every failover capability in every component. This chapter describes the availability and failover features of various components in the Oracle Internet Directory technology stack, and provides guidelines for exploiting them optimally for typical directory deployment.

Oracle Internet Directory and Oracle9*i* Technology Stack

Figure 21–1 gives an overview of the various components of the Oracle Internet Directory stack. Stack communication between separate computers occurs by passing information from one node to the other through several layers of code. Information descends through layers on the client side. It is then packaged for transport across a network medium. The information then proceeds up the stack on the server side where it is translated and understood by the corresponding layers.

Figure 21–1 Oracle Internet Directory/Oracle9i Technology Stack



You can build sufficient fault tolerance mechanisms into each of the layers to ensure maximum availability of the product. In the following sections we describe some of the high availability options available to our customers in each of the layers shown above.

Failover Options on Clients

Incorporating enough intelligence in the clients so that they can failover to alternate Oracle directory servers in case the primary Oracle directory server fails is a good option in some cases. This requires the clients to cache alternate server information and use it upon recognizing connectivity loss. This method of guaranteeing availability is viable only for deployments in which one has full control over the type of clients accessing the directory.

This section contains these topics:

- Alternate Server List from User Input
- Alternate Server List from the Oracle Internet Directory Server

Alternate Server List from User Input

The clients can be designed to take input from the user on the list of alternate Oracle directory servers so that the clients can automatically failover in the event of a failure of the primary server. However, as the number of clients increases, this option would not scale very well in terms of administration of client installations.

Alternate Server List from the Oracle Internet Directory Server

Oracle Internet Directory supports a DSE root attribute called `AltServer`. This is an LDAP Version 3 standard attribute and is to be maintained by the directory administrator. It is expected to have references to other Oracle directory servers in the system with the same set of naming contexts as that of the local server. When connectivity to the local server is lost, clients have the option of accessing one of the servers listed in this attribute. This option requires explicit administrative action to maintain this attribute.

See Also:

- "Managing Attributes by Using Oracle Directory Manager" on page 6-17 and "Managing Attributes by Using Command-Line Tools" on page 6-29 to set the `AltServer` attribute

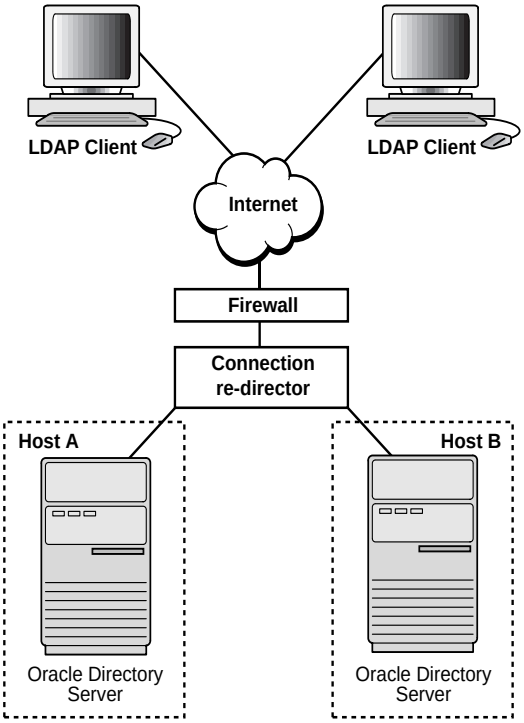
Failover Options in the Public Network Infrastructure

The network used to access Oracle Internet Directory services is called the Public Network Infrastructure. Providing network level load balancing and failover measures (connection re-direction) in the Public Network Infrastructure are highly recommended since these measures provide a high degree of flexibility and transparency to the application clients.

If the Oracle Internet Directory services are accessed from the Internet, this would include a couple of high speed links (T1 to T3) and an intelligent TCP/IP level connection re-director. If the Oracle Internet Directory services are accessed from an Intranet, this would include high speed LAN connections to the server computers running the Oracle directory server and an intelligent TCP/IP level connection re-director. In both cases, there would be more than one computer serving LDAP requests so that failure of one Oracle directory server computer would not affect availability.

Figure 21–2 illustrates a typical Internet deployment of Oracle Internet Directory with network-level failover enabled.

Figure 21–2 Network-Level Failover



In Figure 21–2, the Oracle directory servers (OiD LDAP Servers) can be connected to either the same back-end database or different back-end databases. In this deployment, network-level connection redirection can be accomplished by both hardware and software solutions.

This section contains these topics:

- Hardware-Based Connection Redirection
- Software-Based Connection Redirection

Hardware-Based Connection Redirection

Hardware-based connection redirection technology is available from several vendors. These redirection devices connect directly to the Internet and can route requests among several server computers. They can also detect computer failures and stop routing requests to the failed computer. This feature guarantees that new connections from clients will not be routed to a failed computer. When a computer comes back, the device detects it and starts routing new requests to it. These devices also perform some load balancing, which makes sure that client requests are uniformly distributed.

Some of the vendors providing hardware based re-direction technologies are:

- Accelar Server Switches from Nortel Networks
- Local Director from Cisco
- BIG/ip from F5 Labs Inc.
- Hydra from HydraWEB Technologies
- Equalizer from Coyote Point Systems

Software-Based Connection Redirection

The software-based solutions essentially work in the same manner as their hardware counterparts. Some of the currently available solutions include Dispatch from Resonate and Network Dispatcher from IBM.

Availability and Failover Capabilities in Oracle Internet Directory

Multimaster replication makes it possible for the directory system to be available for both access and updates at all times, as long as at least one of the nodes in the system is available. When a node comes back online after a period of unavailability, replication from the existing nodes will resume automatically and cause its contents to be synchronized transparently.

Any directory system with high availability requirements should always employ a network of replicated nodes in multimaster configuration. A replica node is recommended for each region that is separated from others by a relatively low speed or low bandwidth network segment. Such a configuration, while allowing speedy directory access to the clients in the same region, also serves as a failover arrangement during regional failures elsewhere.

Failover Options in the Private Network Infrastructure

The Private Network Infrastructure is the network used by Oracle Internet Directory and its back-end components to communicate with each other. In cases where Oracle Internet Directory is deployed on the Internet, Oracle Corporation recommends that this network be physically different from the network used to serve client requests. In cases where Oracle Internet Directory is deployed over an Intranet, the same LAN may be used, but Oracle Internet Directory components should have dedicated bandwidth with the help of a network switch. Because Oracle Internet Directory depends on the Private Network Infrastructure for its communications, you must take adequate precautions to guarantee availability in the event of failures in the Private Network. Some of the options available in this area are:

- IP Address Takeover (IPAT)
- Redundant Links

IP Address Takeover (IPAT)

IP address takeover feature is available on many commercial clusters. This feature protects an installation against failures of the Network Interface Cards (NICs). In order to make this mechanism work, installations must have two NICs for each IP address assigned to a server. Both the NICs must be connected to the same physical network. One NIC is always active while the other is in a standby mode. The moment the system detects a problem with the main adapter, it immediately fails over to the standby NIC. Ongoing TCP/IP connections are not disturbed and as a result clients do not notice any downtime on the server.

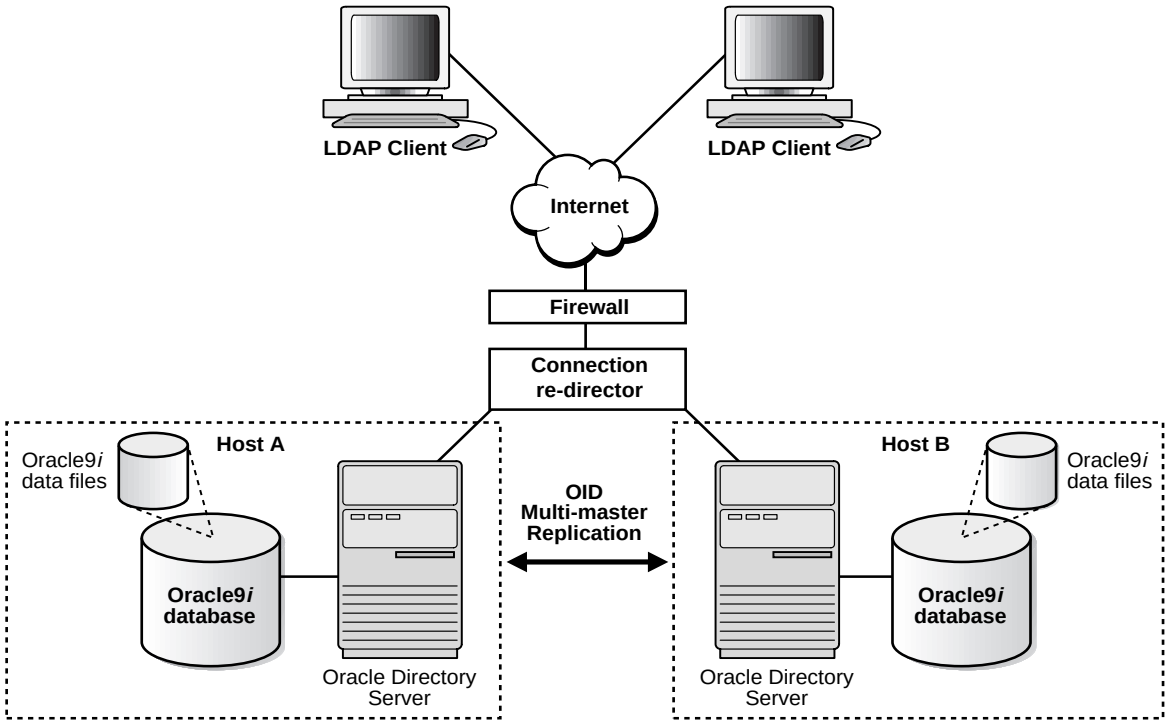
Redundant Links

Since all networks (with the exception of wireless networks) are comprised of wires going from one location to the other, there is a distinct possibility that someone might unintentionally disconnect a wire that is used to link a client computer to a server computer. If you want to take such precautions, use NICs and hubs/switches that come with the capability to use redundant links in case of a link level failure.

High Availability Deployment Examples

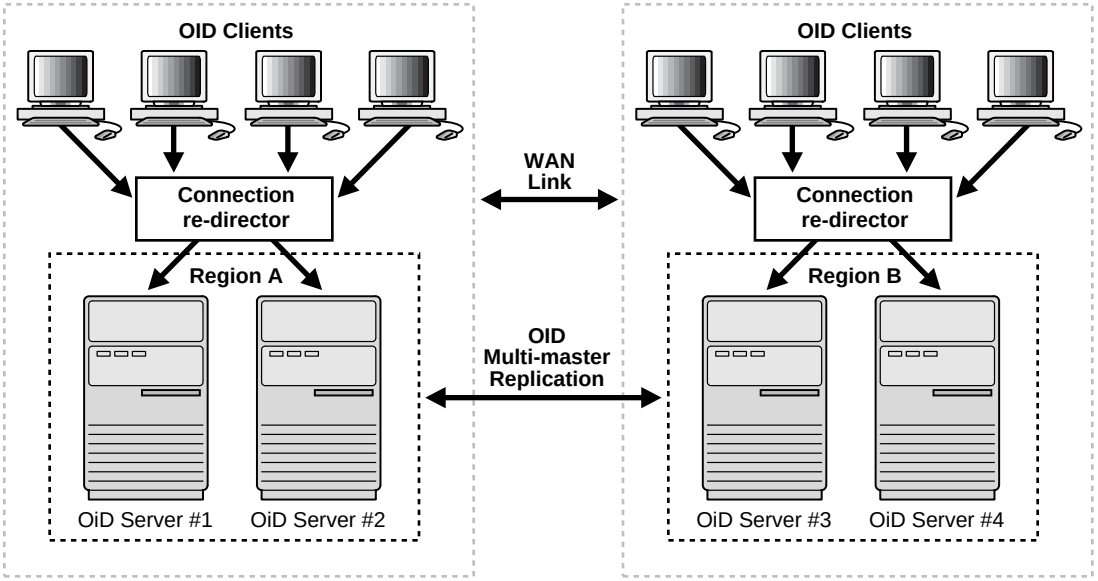
In Figure 21–3, the database and Oracle directory server (OiD LDAP Server) are co-resident on the same computer. Changes made on one directory server instance are reflected on the second directory server instance through multimaster replication. When a failure of the directory server or database server on a particular node occurs, it is elevated to a computer failure so that the connection redirector will stop handing off connections to the computer on which there was a failure.

Figure 21–3 Deployment Example (Two Oracle Internet Directory Nodes in Replication)



As Figure 21–4 illustrates, each of the regions can be set up with two Oracle Internet Directory nodes replicating between each other. This configuration is typical of global directory networks deployed by large enterprises where each of the regions above could potentially represent a continent or a country.

Figure 21–4 Deployment Example 2



Part V

Directory Replication

This part provides detailed discussions of replication and how to manage it. It contains these chapters:

22

Directory Replication Concepts

In "Distributed Directories" on page 2-22, you saw an overview of replication. This chapter provides a closer look. It contains these topics:

- Directory Replication Groups and Replication Agreements
- Oracle9i Replication
- Replication Architecture
- Change Log Purging
- Conflict Resolution in Replication
- The Replication Process

See Also:

- "Replication" on page 2-22 for a more general, conceptual discussion of replication
- Chapter 23, "Oracle Directory Replication Server Administration" for information on managing replication

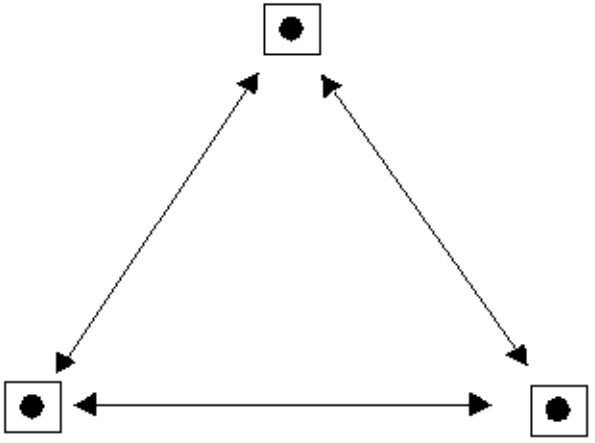
Directory Replication Groups and Replication Agreements

The set of directory servers that participate in replication of a given naming context is called a directory replication group (DRG). A special directory entry, called a replication agreement, represents the replication relationship among the directory servers in a DRG.

It is possible for a directory server to be both a supplier and a consumer of change log information. Oracle Internet Directory uses this feature to support multimaster replication.

Figure 22–1 illustrates a directory replication group in which three nodes share updates with each other in a replication agreement.

Figure 22–1 Directory Replication Group



In Figure 22–1, each bullet represents a node of Oracle Internet Directory. The agreement is identical on each node except for local options such as partitioned naming contexts on the local directory server. The replication agreement on each node lists all the other nodes to which it delivers, and from which it receives, changes.

> **See Also:** "Managing Replication" on page 23-12 for information about how to configure replication agreements

Oracle9*i* Replication

Transport of update information between nodes in a replication agreement is managed by Oracle9*i* Replication, a store-and-forward transport feature available in Oracle9*i*. It allows database tables to be kept synchronized across two Oracle databases.

Oracle9*i* Replication stores local changes and periodically propagates them in batches to consumer servers. The consumer replication servers apply the remote changes to the local directory server and then purge the applied remote changes from their local stores.

Oracle9*i* Replication environments allow read and update access to directory tables anywhere in the Oracle9*i* replication group. Typical Oracle9*i* Replication configurations use row-level replication with asynchronous data propagation.

Oracle9*i* Replication provides proven network tolerance and its data transfer can be controlled and monitored by Oracle Enterprise Manager. Such manageability allows a high degree of flexibility in how the data transfer is scheduled.

> **See Also:** *Oracle9i Replication* in the Oracle Database Documentation Library for information about Oracle9*i* Replication

Replication Architecture

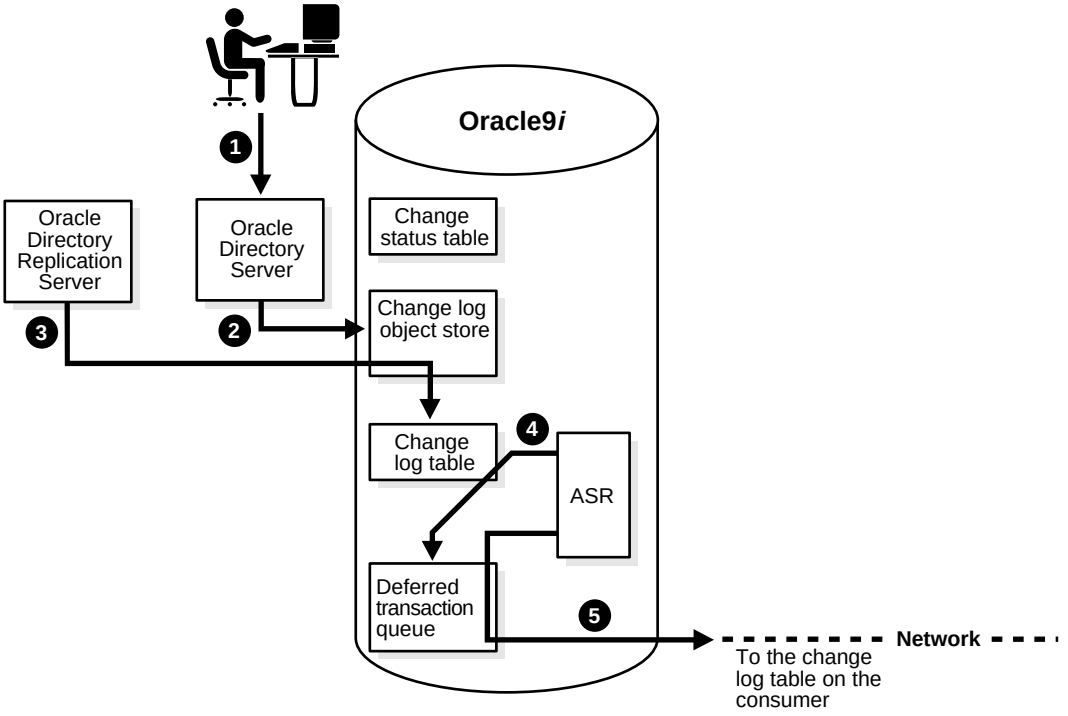
Supplier servers write their changes to change logs, and then regularly send batched directory changes to other consumer servers. Consumer servers receive the change log data, then reproduce the changes locally.

When you configure replication, you specify which nodes in a replication group share changes. Regardless of the number of nodes you introduce into the replication environment, the basic architecture for replication remains the same. Local changes are distributed to remote nodes and applied by replication server processing. To apply the changes on a remote node, the replication server, acting as a client, sends commands to the directory server that implements them.

The rest of this section discusses, in general terms, the replication process, both from the standpoint of the supplier, and from that of the consumer.

The Replication Process on the Supplier Side

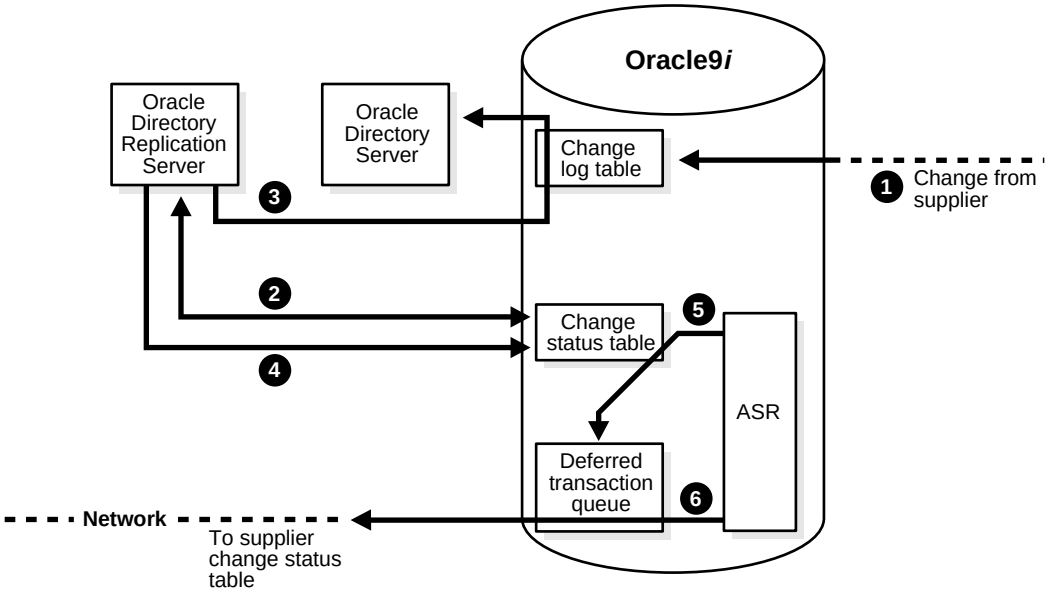
The following graphic and its accompanying text explain what happens on the supplier side during the replication process.



1. An LDAP client issues a directory modification.
2. The Oracle directory server generates a change log object in the change log object store.
3. At a scheduled time, the Oracle directory replication server launches an outbound change log processing thread. This thread translates the change log object into a row—for example, Change entry—in the change log table.
4. When a change entry is committed to the change log table, Oracle9*i* Replication immediately copies the change into the deferred transaction queue.
5. After a scheduled interval, Oracle9*i* Replication pushes pending transactions from the deferred transaction queue across the network to the consumer change log table.

The Replication Process on the Consumer Side

The following graphic and its accompanying text explain the replication process on the consumer side.



1. A change arrives in the consumer change log table from the supplier.
2. The Oracle directory replication server launches a change log processing thread for each supplier, based on a scheduled replication cycle. This thread first consults the change status table for the last change applied from the supplier to the consumer.
3. The Oracle directory replication server then fetches and applies all the new changes from the change log table to the Oracle directory server.
4. The Oracle directory replication server then updates the change status table to record the last change applied from the supplier before exiting.
5. Oracle9*i* Replication copies the change status update into the deferred transaction queue.
6. After the scheduled Oracle9*i* Replication replication interval, Oracle9*i* Replication pushes pending change status updates from the deferred transaction queue to the supplier change status table.

Although, in the previous figures, the roles of supplier and consumer have been separated, in an actual multimaster replication environment, each directory server is both a supplier and a consumer. In such an environment, purging occurs regularly, removing entries that are already applied and entries that have been dropped as candidate changes. Remote change records in the local Changelog table are purged by the garbage collection thread if they have been applied locally. Local change records in the local Changelog table are purged by the garbage collection thread if they have been distributed to all the consumers.

> **See Also:** "Managing Replication" on page 23-12 for information on configuring replication

Change Log Purging

Change log purging takes place in Oracle Internet Directory in two ways:

- Change number-based

  This is the default method. The replication server purges those changes that have already been applied to all the nodes in a DRG.

- Time-based

  You can run this method to augment change number-based purging. To use this additional method, you set a parameter specifying in hours the lifespan of change log objects. For example, you can set this parameter to purge all change log objects that are 24 hours old. Use this method to prevent the change log from becoming too large.

> **See Also:**
>
> - "Viewing and Modifying Replication Configuration Parameters by Using Oracle Directory Manager" on page 23-14
> - "Modifying Replication Configuration Parameters by Using Command-Line Tools" on page 23-15

Conflict Resolution in Replication

Multimaster replication enables updates to multiple directory servers. Conflicts occur whenever the directory replication server attempts to apply remote changes from a supplier to a consumer and fails for some reason.

The following kinds of LDAP operations can lead to conflicts:

- Addition
- Deletion
- Modification
- Modification of either an RDN or a DN

This section contains these topics:

Levels at Which Replication Conflicts Occur

There are two types of conflicts:

- Entry-level conflicts
- Attribute-level conflicts

Entry-Level Conflicts

An entry-level conflicts is caused when the directory replication server attempts to apply a change to the consumer. Such a change could be one of the following types of changes to the consumer:

- Adding an entry that already exists
- Deleting an entry that does not exist
- Modifying an entry that does not exist
- Applying a modifyrdn operation when the DN does not exist

These conflicts can be difficult to resolve. For instance, it may be impossible to resolve a conflict because:

- The entry has been moved to a different location
- The entry has not yet arrived from a supplier
- The entry has been deleted
- The entry never existed on the consumer

If an entry exists and it should not, then it may be because it was added earlier, or that it recently underwent a modifydn operation.

Attribute-Level Conflicts

An attribute-level conflict is caused when two directories are updating the same attribute with different values at different times. If the attribute is single-valued, then the replication process resolves the conflict by examining the timestamps of the changes involved in the conflict.

Typical Causes of Conflicts

Conflicts usually stem from the timing of changes arising from the occasional slowness or transmission failure over wide-area networks. Also, an earlier inconsistency might continue to cause conflicts if it is not resolved in a timely manner.

Automated Resolution of Conflicts

The directory replication server attempts to resolve all conflicts that it encounters by following this process:

1. The conflict is detected when a change is applied.
2. The replication process attempts to reapply the change a specific number of times or repetitively for a specific amount of time after a specific waiting period.
3. If the replication process reaches the retry limit without successfully applying the change, it flags the change as a conflict, which it then tries to resolve. If the conflict cannot be resolved according to the resolution rules (described in the next section), the change is moved to a low-priority, human intervention queue. Changes are then applied according to the time unit specified in the `orclHIQSchedule` parameter in the replication agreement. Before it moves the change, the directory replication server writes the conflict into a log file for the system administrator.

Note: There is no conflict resolution of schema, catalog, and group entries during replication. This is because attempting resolution of such large multi-valued attributes would have a significant negative impact on performance. Be careful to avoid updating such entries from more than one master at a time.

See Also:

- For schema questions, Appendix C, "Schema Elements".
- For catalog questions, the Catalog Management Tool Syntax section in Appendix A, "Syntax for LDIF and Command-Line Tools".
- For group entry questions, the section titled Managing Users, Groups, and Subscribers by Using the Delegated Administration Service on page 9-15.

The Replication Process

This section describes how the automated replication process adds, deletes, and modifies entries, and how it modifies DNs and RDNs. It contains these topics:

- How the Replication Process Adds a New Entry to a Consumer
- How the Replication Process Deletes an Entry
- How the Replication Process Modifies an Entry
- How the Replication Process Modifies a Relative Distinguished Name
- How the Replication Process Modifies a Distinguished Name

How the Replication Process Adds a New Entry to a Consumer

When directory replication server successfully adds a new entry to a consumer, it follows this change application process:

1. The directory replication server looks in the consumer for the DN of the parent of the target entry. Specifically, it does this by looking for a **global unique identifier (GUID)** assigned to the DN of the parent.

2. If the parent entry exists, then the directory replication server composes a DN for the new entry and places the new entry under its parent in the consumer. It then places the change entry in the purge queue.

If the change entry is not successfully applied on the first try:

The directory replication server places the new change entry in the retry queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change entry is not successfully applied on *all but the last* **retry:**

The directory replication server keeps the change entry in the retry queue, decrements the number of retries, and repeats the change application process.

If the change entry is not successfully applied on the last retry:

The directory replication server checks to see if the new entry is a duplicate of an existing entry.

If the change entry is a duplicate entry:

The directory replication server applies the following conflict resolution rules:

* The entry with the older creation time stamp is used.
* If both entries have the same creation time stamp, then the entry with the smaller GUID is used.

If the change entry is used, then the target entry is removed, the change is applied, and the change entry is placed in the purge queue.

If the target entry is used, then the change entry is placed in the purge queue.

If the change entry is not a duplicate entry:

The directory replication server places the change entry in the human intervention queue, and repeats the change application process at the interval you specified in the `orclHIQSchedule` parameter.

If the change entry is not successfully applied after it has been placed in the human intervention queue:

The directory replication server keeps the change in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator. The administrator can use the OID reconciliation tool and the human intervention queue manipulation tool to resolve the conflict.

How the Replication Process Deletes an Entry

When the directory replication server deletes an entry from a consumer, it follows this change application process:

1. The directory replication server looks in the consumer for an entry with a GUID matching the one in the change entry.
2. If the matching entry exists in the consumer, then the directory replication server deletes it. It then places the change entry in the purge queue.

If the change entry is not successfully applied on the first try:

The directory replication server places the change entry in the retry queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change entry is not successfully applied on *all but the last* **retry:**

The directory replication server keeps the change entry in the retry queue, decrements the number of retries, and repeats the change application process.

If the change entry is not successfully applied on the last retry:

The directory replication server places the change entry in the human intervention queue and repeats the change application process at specified intervals.

If the change entry is not successfully applied after it has been placed in the human intervention queue:

The directory replication server keeps the change entry in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator. The administrator can use the OID reconciliation tool and the human intervention queue manipulation tool to resolve the conflict.

How the Replication Process Modifies an Entry

When the directory replication server modifies an entry in a consumer, it follows this change application process:

1. The directory replication server looks in the consumer for an entry with a GUID matching the one in the change entry.
2. If the matching entry exists in the consumer, then the directory replication server compares each attribute in the change entry with each attribute in the target entry.

3. The directory replication server then applies the following conflict resolution rules:
   a. The attribute with the most recent modify time is used.
   b. The attribute with the most recent version of the attribute is used—for example, version 1, 2, or 3.
   c. The modified attribute on the host whose name is closest to the beginning of the alphabet is used.
4. The directory replication server applies the filtered modification, and places the change entry in the purge queue.

If the change entry is not successfully applied on the first try:

The directory replication server places the change entry in the retry queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change entry is not successfully applied on ***all but the last*** **retry:**

The directory replication server keeps the change entry in the retry queue, decrements the number of retries, and repeats the change application process.

If the change entry is ***not*** **successfully applied by the last retry:**

The directory replication server places the change entry in the human intervention queue and repeats the change application process at specified intervals.

If the change entry is not successfully applied after it has been placed in the human intervention queue:

The directory replication server keeps the change entry in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator. The administrator can use the OID reconciliation tool and the human intervention queue manipulation tool to resolve the conflict.

How the Replication Process Modifies a Relative Distinguished Name

When the directory replication server modifies the RDN of an entry in a consumer, it follows this change application process:

1. The directory replication server looks in the consumer for the DN with a GUID that matches the GUID in the change entry.

2. If the matching entry exists in the consumer, then the directory replication server modifies the RDN of that entry and places the change entry in the purge queue.

If the change entry is not successfully applied on the first try:

The directory replication server places the change entry in the retry queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change entry is not successfully applied on *all but the last* **retry:**

The directory replication server keeps the change entry in the retry queue, decrements the number of retries, and repeats the change application process.

If the change entry is not successfully applied on the last retry:

The directory replication server places the change entry in the human intervention queue and checks to see if it is a duplicate of the target entry.

If the change entry is a duplicate entry:

The directory replication server applies the following conflict resolution rules:

* The entry with the older creation time stamp is used.
* If both entries have the same creation time stamp, then the entry with the smaller GUID is used.

If the change entry is used, then the target entry is removed, the change entry is applied, and then placed in the purge queue.

If the target entry is used, then the change entry is placed in the purge queue.

If the change entry is not a duplicate entry:

The directory replication server places the change entry in the human intervention queue, and repeats the change application process at specified intervals.

If the change entry is not successfully applied after it has been placed in the human intervention queue:

The directory replication server keeps the change entry in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator. The administrator can use the OID reconciliation

tool and the human intervention queue manipulation tool to resolve the conflict.

How the Replication Process Modifies a Distinguished Name

When the directory replication server modifies the DN of an entry in a consumer, it follows this change application process:

1. The directory replication server looks in the consumer for the DN with a GUID that matches the GUID in the change entry.

   The directory replication server also looks in the consumer for the parent DN with a GUID that matches the GUID of the new parent specified in the change entry.

2. If both the DN and the parent DN of the target entry exist in the consumer, then the directory replication server modifies the DN of that entry and places the change entry in the purge queue.

If the change entry is not successfully applied on the first try:

The directory replication server places the change entry in the retry queue, sets the number of retries to the configured maximum, and repeats the change application process.

If the change entry is not successfully applied on ***all but the last*** **retry:**

The directory replication server keeps the change entry in the retry queue, decrements the number of retries, and repeats the change application process.

If the change entry is ***not*** **successfully applied by the last retry:**

The directory replication server places the change entry in the human intervention queue and checks to see if it is a duplicate of the target entry.

If the change entry is a duplicate entry:

The directory replication server applies the following conflict resolution rules:

* The entry with the older creation time stamp is used.
* If both entries have the same creation time stamp, then the entry with the smaller GUID is used.

If the change entry is used, then the target entry is removed, the change entry is applied, and then placed in the purge queue.

If the target entry is used, then the change entry is placed in the purge queue.

If the change entry is not a duplicate entry:

The directory replication server places the change entry in the human intervention queue, and repeats the change application process at specified intervals.

If the change entry is not successfully applied after it has been placed in the human intervention queue:

The directory replication server keeps the change entry in this queue, and repeats the change application process at specified intervals while awaiting action by the administrator. The administrator can use the OID reconciliation tool and the human intervention queue manipulation tool to resolve the conflict.

23

Oracle Directory Replication Server Administration

Replication is the mechanism that maintains exact duplicates of specified naming contexts on multiple nodes. This chapter tells you how to install, configure, and manage replication in Oracle Internet Directory.

> **Note:** For Release 9.0.2, you can use Oracle Internet Directory replication only if you have installed **Oracle9i Replication**. This ships with all standalone purchases of Oracle Internet Directory and with Oracle9*i* Enterprise Edition. Oracle9*i* Replication is not included with Oracle9*i* Standard Edition.

This chapter contains these topics:

> **See Also:** "Replication" on page 2-22 for a conceptual discussion of replication

Installing and Configuring Replication

This section describes how to install and initialize directory replication server software on a node.

Each node in a group of directory servers holds an updatable copy, also called an updatable replica, of the same **naming context** or set of naming contexts. These naming contexts are synchronized with each other by replication processing. This group of nodes is called a **directory replication group (DRG)**.

If you are deploying more than one Oracle Internet Directory instance on the same machine, then you cannot uniquely identify each directory server instance by the name of its host. In this case, before installing and configuring replication, follow the instructions in "Identifying a Node as Independent of Its Host" on page 23-31.

To install and configure a replication group, perform these general tasks:

Note:

- The instructions in this section apply to setting up replication in a group of empty nodes. They assume that there is no pre-existing directory data on any of the nodes in the DRG. For instructions on adding a node to an existing DRG, see "Adding a Replication Node" on page 23-22.
- In Oracle Internet Directory Release 9.0.2, procedures and tools are not available to create an environment (directory network) consisting of more than one DRG.
- The directory replication server does not always preserve the spaces between RDN components in the DN during entry replication. In some rare cases, it may not preserve the case of the letters in the DN.
- DSE root-specific data, server configuration data, and replication agreement data are not included in the data replicated between servers in a directory replication group.

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

Note that the typical installation of the Oracle9*i* Enterprise Edition, which is required for the Oracle Internet Directory, includes **Oracle9i Replication**. By contrast, a typical installation of Oracle9*i* Standard Edition does not include Oracle9*i* Replication.

Note: During installation, be sure that each Oracle Internet Directory database instance name is unique on each machine.

See Also: Installation documentation for Oracle Internet Directory

Task 2: Decide Which Node Will Serve as the Oracle9*i* Replication Master Definition Site (MDS)

A **master definition site (MDS)** is any of the Oracle Internet Directory databases in which the administrator is going to run the configuration scripts. A **remote master site (RMS)** is any site other than the MDS that participates in Oracle9*i* Replication.

You must be able to use **Oracle Net Services** to connect to the MDS database and all other nodes that constitute the DRG.

Task 3: Set Up Oracle9*i* Replication for a Directory Replication Group

The following sections lead you through installing and configuring Oracle9*i* Replication through Oracle Internet Directory installation scripts. More advanced Oracle9*i* Replication users may prefer to configure Oracle9*i* Replication through the Oracle9*i* Replication Manager Tool.

> **See Also:** *Oracle9i Replication* in the Oracle Database Documentation Library, and the online help for Oracle9*i* Replication Manager, for information on configuring Oracle9*i* Replication by using the Oracle9*i* Replication Manager

To configure the Oracle9*i* Replication environment to establish a directory replication group (DRG), perform the tasks discussed in these topics:

- On All Nodes, Prepare the Oracle Net Services Environment for Replication
- From the MDS, Configure Oracle9i Replication For Directory Replication

On All Nodes, Prepare the Oracle Net Services Environment for Replication

Follow these steps, described more fully below, on *all nodes* in the directory replication group to prepare the Oracle Net Services environment:

1. Configure sqlnet.ora.
2. Configure tnsnames.ora.
3. Optional: Create rollback table space and rollback segments.
4. Required only if you created rollback table space and rollback segments.
5. Stop and restart the listener.
6. Required only if you created rollback table space and rollback segments.
7. VITAL: Test Oracle Net connections to all nodes from each node in the DRG.

To prepare the Oracle Net Services environment for replication:

1. Configure `sqlnet.ora`.

   The `sqlnet.ora` file should contain the following parameters at minimum:

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

   On UNIX, this file is in $*ORACLE_HOME*`/network/admin`

   On Windows NT, this file is in *ORACLE_HOME*`\network\admin`

2. Configure `tnsnames.ora`.

   Define all Oracle Internet Directory database instances in the DRG on all nodes in the DRG. The `tnsnames.ora` file must contain **connect descriptor** information in the following format for all Oracle Internet Directory databases:

   ```
   net_service_name =
       (DESCRIPTION =
         (ADDRESS =
            (PROTOCOL = TCP)
            (HOST = HOST_NAME_OR_IP_ADDRESS)
            (PORT = 1521))
         (CONNECT_DATA =
            (service_name = service_name)))
   ```

   On UNIX, this file is in $*ORACLE_HOME*`/network/admin`

   On Windows NT, this file is in *ORACLE_HOME*`\network\admin`

   Note: You must domain-qualify the net service name (for example, `sales.com`), but be sure that the domain component matches the one specified in the NAMES.DEFAULT_DOMAIN parameter in the `sqlnet.ora` file.

3. Optional: Create rollback table space and rollback segments.

   You may want to create multiple rollback segments. You can increase the size of the table spaces and segments to meet your system requirements.

   a. Create a tablespace for rollback segments.

      Execute SQL*Plus by typing the following command:

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

At the SQL*Plus prompt, type:

```
CREATE TABLESPACE table_space_name
datafile file_name_with_full_path SIZE 50M REUSE AUTOEXTEND ON NEXT
10M MAXSIZE max_bulk_update transaction_size ex:500M;
```

b. Create rollback segments.

At the SQL*Plus prompt, type the following lines for each rollback segment:

```
CREATE ROLLBACK SEGMENT rollback_segment_name
tablespace table_space_name storage (INITIAL 1M NEXT 1M OPTIMAL 2M
MAXEXTENTS UNLIMITED);
```

Repeat the `CREATE ROLLBACK SEGMENT` command for each rollback segment entered in the initialization parameter file.

4. Required only if you created rollback table space and rollback segments.

Modify the parameters in the initialization parameter file, `init.ora`.

Type the following lines in the initialization parameter file:

```
rollback_segments = (rollback_segment_name_1, rollback_segment_name_2 ...)
SHARED_POOL_SIZE = 20000000
```

Ensure that the total **System Global Area (SGA)** does not exceed 50% of your system's physical memory.

5. Stop and restart the listener.

To stop the listener for the Oracle Internet Directory database, use the listener control utility (lsnrctl). Type the following command at the LSNRCTL command prompt:

```
SET PASSWORD password
STOP [listener_name]
```

`SET PASSWORD` is required only if the password is set in the `listener.ora` file. The password defaults to ORACLE. The default listener name is LISTENER.

To restart the listener for the Oracle Internet Directory database, type the following command at the LSNRCTL command prompt:

```
START [listener_name]
```

6. Required only if you created rollback table space and rollback segments.

Stop and restart the Oracle Internet Directory database.

To stop and restart the Oracle Internet Directory database, you can use SQL*Plus.

See Also:

- *Oracle9i Net Services Administrator's Guide* in the Oracle Database Documentation Library
- *Oracle9i Database Administrator's Guide* in the Oracle Database Documentation Library for instructions on stopping and restarting the database

7. **VITAL:** Test Oracle Net connections to all nodes from each node in the DRG.

   Use SQL*Plus. Test both internal*@net_service_name* and internal*@net_service_name.domain*. If this does not work, then replication will not work.

From the MDS, Configure Oracle9*i* Replication For Directory Replication

To configure Oracle9*i* Replication for the replication group, complete the following steps *from the MDS*:

1. Log on as the Oracle Internet Directory software owner account from a UNIX prompt.
2. Change to the following directory:
   - On UNIX: $*ORACLE_HOME*/ldap/bin
   - On Windows NT: *ORACLE_HOME*\ldap\bin

Note: Before proceeding to the next step, connect as the system user on all nodes, including the MDS, from the MDS console. Ensure the following:

- The Oracle Internet Directory database is up and running
- The Oracle Internet Directory listener is up and running
- The connect descriptor is correct
- The system password is correct

---

3. From the MDS, verify that all Oracle Internet Directory database instances and listeners are running on all nodes in the DRG.
4. From the MDS, at the command prompt, run the following script if the prerequisites in the Note are met:

   ```
   ldaprepl.sh -asrsetup
   ```

   Note:

   - On UNIX, before running this script in the command shell, set the $*ORACLE_HOME* environment variable.
   - On Windows NT, you can run this script only if the MKS Toolkit or the Cygwin UNIX emulation tool is installed.

   The ldaprepl.sh script executes a number of operations.

   - It configures the MDS.
   - It configures the remote master sites.
   - It configures replication push jobs at all sites.
   - It resumes replication at the MDS.
   - It verifies that all steps have completed successfully.
   - It configures the default replication agreement on all nodes.

   As the script runs, it asks for the information in the following table, first for the MDS.

Requested Information	Definition
MDS Global Name	Net service name of the MDS database, as listed in the file `tnsnames.ora`
System password for MDS	System password for the Master Definition Site

   After you have provided this information for the MDS, the script asks you for the global names and system passwords of any other master sites.

   After you have provided the necessary information for the other master sites, the script asks you for the replication administration password. This enables it to create the database account for the replication administrator on all the nodes.

The replication administrator needs this password later when adding or deleting nodes.

Once you have identified all sites, the script shows a table of the information you have provided, and asks for confirmation. If it is not correct, then press N. The script then starts again at the beginning, asking about the MDS again.

After you have provided all the information, the script asks you to verify the information. If the information is correct and you press Y, then the script begins configuring the sites.

This process may take a long time, depending on your system resources and the number of nodes in your DRG. The script keeps you informed of its progress.

Note: If you interrupt the process before it is complete, then you must start at the beginning. Interrupting the process does not negatively affect your re-installation.

If errors arise, see Troubleshooting Replication Setup on page 23-33.

See Also:

- *Oracle9i Database Administrator's Guide* in the Oracle Database Documentation Library for instructions on ensuring that the database and listener are running
- *Oracle9i Net Services Administrator's Guide* in the Oracle Database Documentation Library for instructions on ensuring that the connect string is correct

Task 4: Load Data into the Directory

If you have a small number of entries to add to the DRG, you can wait until you have completely configured the DRG, then use ldapadd to load the data to one of the nodes. The entries will then be replicated to the other nodes at the specified time.

If you have a large amount of data to load into the DRG, then use the bulkload utility. To do this:

1. On any of the nodes, enter:

   ```
   bulkload.sh -connect net_service_name -check -generate file_with_absolute_path_name
   ```

2. From the same node, enter:

   ```
   bulkload.sh -connect net_service_name_1 -load
   ```

3. Repeat Step 2, each time replacing *net_service_name_1* with the net service name of another node in the DRG, until you have loaded the data onto all the nodes. For example, enter:

   ```
   bulkload.sh -connect net_service_name_2 -load
   ```

   then enter:

   ```
   bulkload.sh -connect net_service_name_3 -load
   ```

   and so on, until you have bulkloaded the data onto each node in the DRG.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

See Also:

- See "bulkload Syntax" on page A-35 for syntax and usage notes.

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

To start Oracle directory server instances on all nodes, run the following command:

```
oidctl connect=net_service_name server=oidldapd instance=instance_number_of_
ldap_server flags='-p port' start
```

Be sure that the change logging option for the directory server is set to the default, namely, TRUE.

Note: The `instance_number_of_ldap_server` need not be unique across the entire DRG. For example, you can have `instance=1` on both node A on node B.

See Also: Chapter 3, "Preliminary Tasks and Information" for more information on starting an Oracle directory server **instance.**

Task 6: Start the Replication Servers on All Nodes in the DRG

To start replication servers on all nodes, type the following command:

```
oidctl connect=net_service_name server=oidrepld instance=1
   flags='-h host_name_of_this_computer -p port' start
```

Note that the instance number does not need to be unique across the entire DRG.

See Also: Chapter 5, "Oracle Directory Server Administration" for information on starting the replication servers

You can turn off the multimaster flag, which occurs in the directory replication server, by changing the value of the `-m` flag in the OID Control Utility command for Oracle directory server from the default, namely, TRUE, to FALSE. This is useful for reducing performance overhead if you are deploying a single master with read-only replica consumers. The multimaster option controls conflict resolution, which serves no purpose if you are deploying a single master.

See Also: "Conflict Resolution in Replication" on page 22-7

Note: As part of Task 3, the ldaprepl script set normal defaults enabling you to simply start the replication servers. If you wish to alter these defaults, see Managing Replication on page 23-12.

Task 7: Test Directory Replication

Use Oracle Directory Manager to verify that the directory replication servers are running, then test directory replication by doing the following:

1. Log in to Oracle Directory Manager as orcladmin.
2. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Entry Management.
3. Create a single entry on the MDS node.

   The identical entry appears in approximately 1 to 10 minutes on the RMS. You can adjust the timing in the replication server configuration set entry. If entries are modified on any nodes in the DRG, then the changes will be replicated.

Managing Replication

Once you have installed and configured replication, you have the option of modifying the default parameters for server configuration and replication agreements. Replication agreements are entries that list the member nodes (in a replication group) that share their changes. Replication agreements are referenced by replication server configuration parameters that load when the directory replication server runs.

Directory replication server configuration parameters are stored as special attributes in directory entries. You can configure replication parameters and replication agreements the same way you configure the Oracle Internet Directory. You can do either of the following:

- Use Oracle Directory Manager to view and modify configuration entries and agreement entries, as described on page 23-14 and on page 23-18,

or

- Use command-line tools, such as ldapadd and ldapmodify, to alter the configuration and agreement entries, as described on page 23-15 and on page 23-19.

Note: No change to any configuration parameter or replication agreement takes effect until the replication server is restarted.

This section explains both approaches and contains the following topics:

- Modifying Directory Replication Server Configuration Parameters
- Modifying Replication Agreement Parameters
- Changing the Replication Administrator's Password on All Nodes

Modifying Directory Replication Server Configuration Parameters

The directory replication server configuration parameters are stored in the replication server **configuration set entry**, which has the following DN:

```
cn=configset0,cn=osdrepld,cn=subconfigsubentry
```

This entry contains replication attributes that control replication processing. You can modify some of these attributes. Note that the `orclDirReplGroupAgreement` attribute contains a replication agreement identifier. In this release, only one replication agreement is possible.

Table 23–1 lists and describes the directory replication server configuration parameters.

Table 23–1 Directory Replication Server Configuration Parameters

Parameter name	Description	Default Values	Modifiable?
modifyTimestamp	Time of entry creation or modification		No
modifiersName	Name of person creating or modifying the entry		No
orclChangeRetryCount	Single-valued attribute. The number of processing retry attempts for a change-entry before being moved to the human intervention queue. The value for this parameter must be equal to or greater than 1 (one).	10	Yes

Table 23–1 Directory Replication Server Configuration Parameters

Parameter name	Description	Default Values	Modifiable?
orclPurgeSchedule	Single-valued attribute. Specifies purge (garbage collection) interval in minutes. Removes entries that are already applied or have been dropped as candidate changes. This thread is initiated periodically based on the frequency that you set. The value for this parameter must be equal to or greater than 1 (one).	10 minutes	Yes
orclThreadsPerSupplier	Number of worker threads directory replication server provides for each supplier for change log processing. The value for this parameter must be equal to or greater than 1 (one).	5	Yes
orclDirReplGroupAgreement	Multi-valued attribute. Identifies the symmetrical replication agreements for which this server is responsible.	orclagreementid=000001, cn=orclreplagreements	No
orclChangeLogLife	Single-valued attribute. Specifies in hours the time for the life of entries in the change log store. 0 (zero) indicates that this is a change number-based purge. **See Also:** "Change Log Purging" on page 22-6	0	Yes

Viewing and Modifying Replication Configuration Parameters by Using Oracle Directory Manager

To view and modify replication configuration parameters:

1. In the navigator pane, expand Oracle Internet Directory > *directory_server_instance* > Server Management > Replication Server.
2. Select the replication configuration set whose parameters you want to view or modify. The corresponding tab pages appear in the right pane.

Note: No change to any configuration parameter or replication agreement takes effect until the replication server is restarted.

Configuration parameters appear in the General tab page. Use this tab page to view replication configuration parameters, and modify many of them. The following table describes the fields in this tab page.

Field	Description
Modify Timestamp	Time of entry creation or modification in **UTC (Coordinated Universal Time)**. You cannot modify this parameter.
Modifier's Name	Name of person creating or modifying the entry. You cannot modify this parameter.
Change Retry Count	Type the number of attempts that the conflict resolution process tries to apply each update before giving up and logging the incident. The default is 10.
Purge Schedule	Type the number of minutes in between garbage collections. The replication garbage collection thread removes entries that are already applied or have been dropped as candidate changes. The default is 10.
Number of Threads Per Supplier	Type the number of worker threads the directory replication server provides for each supplier for change log processing. The default is 5.
Set	Type the configuration identifier.
Change Log Life	Type the number of hours for the life of the change log objects. **See Also:** "Change Log Purging" on page 22-6

Modifying Replication Configuration Parameters by Using Command-Line Tools

To modify replication configuration parameters by using command-line tools, use the syntax documented in "ldapmodify Syntax" on page A-15.

Modifying the Garbage Collection Interval by Using ldapmodify This example uses an input file named `mod.ldif` to change the garbage collection interval from the default of 10 minutes to 30 minutes.

1. Edit `mod.ldif` as follows:

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

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

```
ldapmodify -h my_host -p 389 -f mod.ldif
```

3. Restart the directory replication server.

Modifying the Change Log Life Parameter by Using ldapmodify This example uses an input file named `mod.ldif` to change the change log life parameter to 10 hours:

1. Edit `mod.ldif` as follows:

```
dn: cn=configset0,cn=oidrepld,cn=subconfigsubentry
changetype: modify
replace: orclChangeLogLife
orclChangeLogLife: 10
```

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

```
ldapmodify -h my_host -p 389 -f mod.ldif
```

3. Restart the directory replication server.

Modifying the Number of Retries Before a Change Is Moved into the Purge Queue by Using ldapmodify This example uses an input file named `mod.ldif` to change the number of retry attempts from the default of ten times to five times. Specifically, after attempting to apply an update five times, the update is dropped and logged in the replication log.

1. Edit `mod.ldif` as follows:

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

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

```
ldapmodify -h my_host -p 389 -f mod.ldif
```

3. Restart the directory replication server.

Modifying the Number of Worker Threads Used in Change Log Processing by Using ldapmodify This example uses an input file named `mod.ldif` to change the number of worker threads used in change log processing to 7:

1. Edit `mod.ldif` as follows:

```
dn: cn=configset0,cn=osdrepld,cn=subconfigsubentry
changetype: modify
replace: orclthreadspersupplier
orclthreadspersupplier: 7
```

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

```
ldapmodify -h my_host -p 389 -f mod.ldif
```

3. Restart the directory replication server.

> **See Also:** "Restarting Directory Server Instances" on page 3-7 for instructions on restarting the directory replication server

Modifying Replication Agreement Parameters

Replication agreement parameters are stored in the replication agreement entries which have the following DN:

`orclAgreementID=`*`id number`*`,cn=orclreplagreements`

This entry contains attributes that pertain only to the nodes participating in this agreement. You can create multiple replication agreements to manage replication between reciprocating nodes, but you can reference only one of them in your start-server message by using Oracle Directory Manager. For Oracle Internet Directory ~~Release 9.0.2~~, only one replication agreement can be used.

In the parameter `DirectoryReplicationGroupDSAs`, type the host names for all of the nodes in the DRG. This list must be identical on all the nodes (see Identifying a Node as Independent of Its Host on page 23-31.

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

See Also:

- "Viewing and Modifying Replication Agreement Parameters by Using Oracle Directory Manager" on page 23-18
- "Modifying Replication Agreement Parameters by Using ldapmodify" on page 23-19

Viewing and Modifying Replication Agreement Parameters by Using Oracle Directory Manager

To view and modify replication agreement parameters by using Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Server Management > Replication Server, and select Default Configuration Set.
2. In the right pane, select the Agreement tab to display the replication agreement.

   The fields in this tab page are described in the following table. You can view the parameters and modify some of them by double-clicking the attributes.

Field	Description	Default Values	Modifiable?
Agreements ID	Unique identifier for a replication agreement.	000001	No
Excluded Naming Contexts	Multivalued attribute. Specifies naming contexts excluded from this replication agreement. Changes to entries in these naming contexts sent from other replicas are not applied on the local node.	None	Yes
Replication Group Nodes	Multi-valued attribute. Specifies nodes participating in symmetrical replication agreement. *Nodes that you specify here share updates with one another.*		Yes
Update Schedule	Replication update interval for new changes and those being retried. The value is in minutes.	1	Yes

Field	Description	Default Values	Modifiable?
Orcl HIQSchedule	Replication update interval for the human intervention queue. The value is in minutes. The value is typically higher that orclUpdateSchedule. This gives administrators time to change the DIT structures when retrying an update fails to resolve a conflict.	10	Yes
Replication Protocol	Specifies the replication protocol used in this replication agreement. The supported protocol is Oracle9*i* Replication.	ODS_ASR_1.0	No

Note: Be sure to add all host names for all nodes in the DRG into the Replication Group Nodes field. Do this for all nodes in the DRG.

3. If you want to return to the values that appeared when you first opened this pane, then click Revert. If you are satisfied with your changes, then click Apply.

Modifying Replication Agreement Parameters by Using ldapmodify

The following table lists and describes the replication agreement parameters.

Parameter	Description	Default Values	Modifiable?
`orclAgreementID`	Unique identifier for a replication agreement.	000001	No
`orclExcludedNamingcontexts`	Multi-valued attribute. Specifies naming contexts excluded from this replication agreement. Changes to entries in these naming contexts sent from other replicas are not applied on the local node.	None	Yes

Parameter	Description	Default Values	Modifiable?
orclDirReplGroupDSAs	Multi-valued attribute. Specifies nodes participating in symmetrical replication agreement. *Nodes that you specify here share updates with one another.*		Yes
orclUpdateSchedule	Replication update interval for new changes and those being retried. The value is in minutes.	1	Yes
OrclHIQSchedule	Replication update interval for the human intervention queue. The value is in minutes. The value is typically higher that orclUpdateSchedule. This gives administrators time to change the DIT structures when retrying an update fails to resolve a conflict.	10	Yes
orclReplicationProtocol	Specifies the replication protocol used in this replication agreement. The supported protocol is Oracle9*i* Replication.	ODS_ ASR_1.0	No

To add more nodes to the values in a replication agreement entry, run ldapmodify at the command line, referencing an LDIF-formatted file.

This example uses an input file named `mod.ldif` to add two nodes to a replication agreement:

1. Edit `mod.ldif` as follows:

```
dn: orclagreementid=000001,cn=orclreplagreements
changetype: modify
add: orcldirreplgroupdsas
orcldirreplgroupdsas: hollis
orcldirreplgroupdsas: eastsun-11
```

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

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

3. Restart the directory replication server.

This procedure modifies the entry containing the replication agreement whose DN is `orclagreementid=000001,cn=orclreplagreements`. The input file adds the two nodes, `hollis` and `eastsun-11`, into the replication group governed by `oraclagreementid 000001`.

Note: You must include the new nodes—for example, `hollis` and `eastsun-11` in the above sample LDIF file—in the `orclDirReplGroupDSAs` parameter on each node in the replicated environment before you start the replication process.

"Adding a Replication Node" on page 23-22 explains the process of adding a new node to a replication environment.

Because Oracle Internet Directory Release 9.0.2 supports only one configuration set for directory replication server, you do not need to specify a configuration set.

Changing the Replication Administrator's Password on All Nodes

The Oracle9*i* Replication administrator can change the password for administering Oracle9*i* Replication on all nodes by using the -chgpaswd utility. To launch this utility, enter:

```
ldaprepl.sh -chgpswd
```

The -chgpswd utility prompts you for the MDS Global Name—that is, the name of the Master Definition Site—the current password, and the new password. It then asks you to confirm the new password. If you enter an incorrect current password, then the -chgpswd utility asks you to enter it again, up to three times.

Adding a Replication Node

There are two ways to add a new node to a live replication group.

- Using ldifwrite

  This method, described in this section, is the easier of the two. The process can be fully automated, and the generated file can be used for partial replication. Use this procedure unless your directory is very large. Backup using this method can take up to seven hours for a directory with one million entries.

- Using cold backup

  This method, described in Chapter 24, "Addition of a Node by Using the Database Copy Procedure", cannot be fully automated and cannot be reused for partial replication. However, cold backup takes much less time for a large directory server. For example, if your directory has more than a million entries, then use this method.

Before you add a replication node:

- Prepare the Oracle Net Services environment as described in "On All Nodes, Prepare the Oracle Net Services Environment for Replication" on page 23-4.
- Be sure that there is no pre-existing data on the new node. Any pre-existing data will not be replicated to the other participants in the **directory replication group (DRG)**. To replicate pre-existing data:

  1. Extract the data to an LDIF file by using ldapsearch with the `-L` option.
  2. Delete all exported entries from the new node.
  3. After the new node is added to the DRG and can replicate new data to the other nodes, reload the exported data by using ldapadd.

To add a replication node to a functioning DRG of any significant size, follow these general steps, each of which is more fully described later in this chapter.

Task 1: Stop the Directory Replication Server on All Nodes

Task 2: Identify a Sponsor Node and Switch the Sponsor Node to Read-Only Mode

Task 3: Backup the Sponsor Node by Using ldifwrite

Task 4: Perform Oracle9i Replication Add Node Setup

Task 5: Switch the Sponsor Node to Updatable Mode

Task 6: Start the Directory Replication Server on All Nodes Except the New Node

Task 7: Load Data into the New Node by Using bulkload

Task 8: Start LDAP Server on the New Node

Task 9: Start the Directory Replication Server on the New Node

---

Note: Commands shown in the following tasks require the following types of items to be stored as follows:

- Binaries: `$ORACLE_HOME/bin`
- SQL scripts: `$ORACLE_HOME/ldap/admin`
- UNIX scripts: `$ORACLE_HOME/ldap/bin`

Before beginning "Task 1: Install Oracle Internet Directory on All Nodes in the DRG", be sure that all three of these types of items are in the path.

Task 1: Stop the Directory Replication Server on All Nodes

To stop the directory replication server, run the following command on each node in the LDAP replication group:

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

Note: The instance number may not be 1. Check the running process to discover the instance number in use here.

Task 2: Identify a Sponsor Node and Switch the Sponsor Node to Read-Only Mode

A sponsor node is one that will supply the data to the new node. To identify a sponsor node and switch it to read-only mode:

1. Create a new file, `change_mode.ldif`, containing the following:

```
dn:
changetype: modify
replace: orclservermode
orclservermode: r
```

2. Run the following commands against the identified sponsor node:

```
ldapmodify -D "cn=orcladmin" -w welcome -h host_name_of_sponsor_node
-p port -f change_mode.ldif

oidctl connect=net_service_name server=oidldapd restart
```

This restarts all running Oracle directory servers on the sponsor node in Read-Only mode. It takes approximately fifteen seconds for a directory server to restart.

Note: While the sponsor node is in read-only mode, you may not make any updates to it. You may, however, update any of the other nodes, but those updates are not replicated immediately.

Also, the sponsor node and the **MDS** may be the same node.

Task 3: Backup the Sponsor Node by Using ldifwrite

Because this may take a long time, you may start "Task 4: Perform Oracle9i Replication Add Node Setup" while backup is in process.

Enter the following command:

```
ldifwrite -c db_connect_string -b "" -f output_ldif_file
```

Task 4: Perform Oracle9*i* Replication Add Node Setup

You can perform this task at the same time as you are performing "Task 3: Backup the Sponsor Node by Using ldifwrite".

From the sponsor node, run the following script:

```
ldaprepl.sh -addnode
```

This script executes a number of operations.

- It quiesces Oracle9*i* Replication at the sponsor node and any other existing **master site**.
- It configures the master sites and the new node. A master site is any site other than the sponsor node that participates in LDAP replication.
- It configures replication push jobs at all sites including the new node.
- It checks that all steps have completed successfully. (This may take a long time.)
- It performs post-add-node operation.

As the script runs, it asks for the information in Table 23–2, first for the sponsor node then for the existing master sites.

Table 23–2

Requested Information	Definition
MDS Global Name	Net service name of the MDS database, as listed in the file `tnsnames.ora`
System password for MDS	System password for the Master Definition Site

When you have identified all the existing master sites, enter `N`. The script then asks for the global name of the node you want to add, the system password for that node, and the replication administrator's database account password. Once you have provided that information, the script shows you a table of the information you have provided, and asks for confirmation.

If the information is not correct, then press `N`. The script then starts again at the beginning, asking the same information. If the information is correct and you enter `Y`, then the script begins configuring the sites.

This process can take a long time, depending on your system resources and the size of your DRG. The script keeps you informed of its progress.

Note: If for any reason you must interrupt the process before it is complete, then you must start from the beginning.

If errors arise, see Troubleshooting Replication Setup on page 23-33

Task 5: Switch the Sponsor Node to Updatable Mode

To switch the sponsor node to updatable mode:

1. Edit `change_mode.ldif` to the following:

```
dn:
changetype: modify
replace: orclservermode
orclservermode: rw
```

2. Run the following commands on the sponsor node:

```
ldapmodify -D "cn=orcladmin" -w welcome -h host_name_of_sponsor_node
-p  port  -f change_mode.ldif
```

```
oidctl connect=net_service_name server=oidldapd restart
```

Note: Task 6 is very similar to Task 3. The only difference is that the `orclservermode` parameter in `change_mode.ldif` is being set back to `rw`, that is, Read-Write, in this step.

Task 6: Start the Directory Replication Server on All Nodes Except the New Node

To start the directory replication server, type the following command:

```
oidctl connect=db_connection_string server=oidrepld instance=1
flags='-h host -p port' start
```

Verify that no directory or replication processes are running on the new node.

Task 7: Load Data into the New Node by Using bulkload

To load data, type the following command:

```
bulkload.sh -connect db_connect_string_of_new_node -generate -load
-restore absolute_path_to_the_ldif_file_generated_by_ldifwrite
```

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Task 8: Start LDAP Server on the New Node

To start the LDAP server, type the following command:

```
oidctl connect=db_connect_string_of_new_node server=oidldapd
instance=1 flags='-p port' start
```

Task 9: Start the Directory Replication Server on the New Node

> **Note:** If you need to change configuration or agreement parameters, see Managing Replication on page 23-12.

To start the directory replication server, type the following command:

```
oidctl connect=db_connect_string_of_new_node server=oidrepld instance=1
flags='-h host_name_of_new_node -p port' start
```

Deleting a Replication Node

At times, you may want to delete a node from a **DRG**. For example, if the addition of a new node did not fully succeed as a result of system errors, then you need to delete that node.

You can delete a replication node from a DRG only if there are more than two nodes in the DRG.

To delete a replication node from a directory with fewer than a million entries, follow these steps, each of which is more fully described in this section.

Task 1: Stop the Directory Replication Server on All Nodes

Task 2: Stop All Processes in the Node to be Deleted

Task 3: Delete the Node from the Master Definition Site

Task 4: Start the Directory Replication Server on All Nodes

> **Note:** Commands shown in the following steps require that the following variables be stored in the corresponding directories:
>
> - Binaries: $*ORACLE_HOME*/bin
> - SQL scripts: $*ORACLE_HOME*/ldap/admin
> - UNIX scripts: $*ORACLE_HOME*/ldap/bin
>
> Before beginning Task 1, be sure that all three variables are in the path.

Task 1: Stop the Directory Replication Server on All Nodes

To stop the directory replication server, run the following command on each node in the DRG:

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

Note: The instance number may vary.

Task 2: Stop All Processes in the Node to be Deleted

Stop the **OID Control Utility** and the **OID Monitor**.

See Also:

- "Stopping an Oracle Directory Server Instance" on page 3-5 for instructions about stopping the OID Control Utility
- "Stopping the OID Monitor" on page 3-3 for instructions about stopping the OID Monitor

Task 3: Delete the Node from the Master Definition Site

From the **MDS**, run the following script:

```
ldaprepl.sh -delnode
```

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

This script executes these operations:

- It quiesces **Oracle9i Replication** at the MDS and every other existing **master site**.
- It deletes the node from the Directory Replication Group.
- It verifies that all steps have completed successfully.

As the script runs, it asks for the global name of the MDS, the global name of the node you want to delete, and the replication administrator's password. Once you have provided that information, the script shows you a table of the information you have provided, and asks for confirmation. If the information is not correct, then press N. The script then starts again at the beginning, asking the same information. If the information is correct and you enter Y, then the script begins configuring the sites.

This process can take a long time, depending on your system resources and the size of your DRG. The script keeps you informed of its progress.

Note: If, for any reason, you must interrupt the process before it is complete, then you must start from the beginning.

Task 4: Start the Directory Replication Server on All Nodes

To start the directory replication server, type the following command:

```
oidctl connect=net_service_name server=oidrepld instance=1
flags='-h host -p port' start
```

Resolving Conflicts Manually

This section contains these topics:

- Monitoring Replication Change Conflicts
- Examples of Conflict Resolution Messages
- Using the Human Intervention Queue Manipulation Tool
- Using the OID Reconciliation Tool

Monitoring Replication Change Conflicts

If a conflict has been written into the log, then it means that the system is not able to resolve it by following its resolution procedure. To avoid further replication change conflicts arising from earlier unapplied changes, it is important to monitor the logs regularly.

To monitor replication change conflicts, examine the contents of the replication log. You can distinguish between messages by their respective timestamps.

Examples of Conflict Resolution Messages

Conflict resolution messages, examples of which are shown below, are logged in the file `oidrepld00.log`. The path for this file is *`ORACLE_HOME`*`/ldap/log`. The result of each attempt to resolve the replication conflict is displayed at the end of each conflict resolution message.

Example 1: An Attempt to Modify a Non-Existent Entry

```
2000/08/03::10:59:05:  ************ Conflict Resolution Message ************
2000/08/03::10:59:05:  Conflict reason: Attempted to modify a non-existent
entry.
2000/08/03::10:59:05:  Change number:1306.
2000/08/03::10:59:05:  Supplier:eastlab-sun.
2000/08/03::10:59:05:  Change type:Modify.
2000/08/03::10:59:05:  Target
DN:cn=ccc,ou=Recruiting,ou=HR,ou=Americas,o=IMC,c=US.
2000/08/03::10:59:05:  Result: Change moved to low priority queue after failing
on 10th retry.
```

Example 2: An Attempt to Add an Existing Entry

```
2000/08/03::10:59:05:  ************ Conflict Resolution Message ************
2000/08/03::10:59:05:  Conflict reason: Attempted to add an existing entry.
2000/08/03::10:59:05:  Change number:1209.
2000/08/03::10:59:05:  Supplier:eastlab-sun.
2000/08/03::10:59:05:  Change type:Add.
2000/08/03::10:59:05:  Target DN:cn=Lou Smith, ou=Recruiting, ou=HR,
ou=Americas, o=IMC, c=US.
2000/08/03::10:59:05:  Result: Deleted duplicated target entry which was created
later than the change entry. Apply the change entry again.
```

Example 3: An Attempt to Delete a Non-Existent Entry

```
2000/08/03::10:59:06:  ************ Conflict Resolution Message ************
2000/08/03::10:59:06:  Conflict reason: Attempted to delete a non-existent
entry.
2000/08/03::10:59:06:  Change number:1365.
2000/08/03::10:59:06:  Supplier:eastlab-sun.
2000/08/03::10:59:06:  Change type:Delete.
2000/08/03::10:59:06:  Target DN:cn=Lou
Smith,ou=recruiting,ou=hr,ou=americas,o=imc,c=us.
2000/08/03::10:59:06:  Result: Change moved to low priority queue after failing
on 10th retry.
```

Using the Human Intervention Queue Manipulation Tool

The human intervention queue manipulation tool enables you to move the changes from the human intervention queue to either the retry queue or the purge queue. Moving the change to the purge queue means that there are no further attempts to re-apply the changelog entry. Perform the following general steps to address changes in the human intervention queue:

1. Shutdown the directory replication server.
2. Analyze the replication log.
3. Use the human intervention queue manipulation tool to move the changes to either the retry queue or the purge queue as described in the following sections.

See Also: "Human Intervention Queue Manipulation Tool Syntax" on page A-49

Using the OID Reconciliation Tool

When the directory replication server encounters inconsistent data, you can use the OID reconciliation tool to synchronize the entries on the consumer with those on the supplier. When you do this, perform the following general steps:

1. Set the supplier and the consumer to read-only mode.
2. Ensure that the supplier and the consumer are in tranquil state. If they are not in a tranquil state, then wait until they have finished updating.
3. Identify the inconsistent entries or subtree on the consumer.
4. Use the OID reconciliation tool to fix the inconsistent entries or subtree on the consumer.
5. Set the participating supplier and consumer back to read-write mode.

See Also: "OID Reconciliation Tool Syntax" on page A-52 for syntax and an explanation of how OID reconciliation tool works.

Identifying a Node as Independent of Its Host

In most deployments, a node in a DRG is uniquely identified by the name of the host where Oracle Internet Directory is installed. However, when there are multiple installations of Oracle Internet Directory on the same host, the host name cannot be a unique node identifier. In this case, you should use the `orclReplicaId` attribute of the Root DSE.

When you identify a node in a DRG by using `orclReplicaId` instead of the host name, follow the steps in this section.

> **Note:** Do not perform any updates on the nodes in the DRG until you have modified the `orclReplicaId` Root DSE attribute on all the nodes.

1. On each node in the DRG, give the `orclReplicaId` a unique value. For example, if there are three nodes on the same computer, and the corresponding directory servers are running on port1, port2 and port3, then you would perform following modifications:

```
ldapmodify -v -h host -p port1  << EOF
dn:
changetype: modify
replace: orclreplicaid
orclreplicaid : replica001

ldapmodify -v -h host -p port2  << EOF
dn:
changetype: modify
replace: orclreplicaid
orclreplicaid : replica002

ldapmodify -v -h host -p port3  << EOF
dn:
changetype: modify
replace: orclreplicaid
orclreplicaid : replica003
```

2. After you have modified `orclreplicaid` on all the nodes, perform replication setup as described in "Installing and Configuring Replication" on page 23-2.
3. When you modify the DRG as described in "Modifying Replication Agreement Parameters by Using ldapmodify" on page 23-19, give the `orcldirreplgroupdsas` attribute the same value you assigned to orclreplicaid. To use the previous example, you would give the `orcldirreplgroupdsas` attribute the values `replica001`, `replica002`, `replica003`.

Note: Once you have set up replication, do not modify the `orclreplicaId` attribute.

Troubleshooting Replication Setup

If the replication setup fails, then do the following:

1. Check the `$ORACLE_HOME/ldap/admin/logs/ldaprepl.log` file to see the status.
2. Go to the directory `$ORACLE_HOME/ldap/admin` and check the status of replication jobs by running the following command:

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

   Run this command for each node in the DRG. Issuing this command should result in no rows being selected. If rows are selected containing the failed status and error messages, then this means that Oracle9*i* Replication set up failed. In this case, you may:

   - Run the script from the beginning
   - Consult the troubleshooting chapter in *Oracle9i Replication* in the Oracle Database Documentation Library
   - Determine a solution from error message information by consulting an expert in Oracle9*i* Replication

   See Also:

   - From the MDS, Configure Oracle9i Replication For Directory Replication on page 23-7, and
   - Task 4: Perform Oracle9i Replication Add Node Setup on page 23-24

24

Addition of a Node by Using the Database Copy Procedure

This chapter tells how to add a new node to an existing replicating system by using the database copy procedure, also known as **cold backup**.

> **Note:** Because this procedure involves copying Oracle data files, faster performance depends on the underlying network. If the underlying network is weak, then it may be better to implement the method described in Chapter 23, "Oracle Directory Replication Server Administration", 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 chapter contains these topics:

Assumptions

This document assumes that the UNIX directories are created according to Optimal Flexible Architecture (OFA), the set of configuration guidelines for efficient and reliable Oracle databases.

> **See Also:** The Oracle installation guide for your operating system for more information on OFA

Sponsor Directory Site Environment

Set up the environment of the sponsor site. In the example shown throughout this chapter, the host name is rst-sun.

```
Hostname    = rst-sun
ORACLE_BASE = /private/oracle/app/oracle
ORACLE_HOME = /private/oracle/app/oracle/product/8.1.6
ORACLE_SID  = LDAP
LD_LIBRARY_PATH = $ORACLE_HOME/lib
NLS_LANG    = AMERICAN_AMERICA.UTF8
datafile location = /private/oracle/oradata/LDAP
Dump destination =  /private1/oracle/app/oracle/admin/LDAP/pfile,
                    /private1/oracle/app/oracle/admin/LDAP/bdump,
                    /private1/oracle/app/oracle/admin/LDAP/cdump,
                    /private1/oracle/app/oracle/admin/LDAP/udump,
                    /private1/oracle/app/oracle/admin/LDAP/create
```

New Directory Site Environment

Set up the environment for the new directory site. In the example shown throughout this chapter, the new site is on the node named dsm-sun.

```
Hostname = dsm-sun
ORACLE_BASE = /private1/oracle/app/oracle
ORACLE_HOME = /private1/oracle/app/oracle/product/8.1.6
ORACLE_SID  = NLDAP
LD_LIBRARY_PATH = $ORACLE_HOME/lib
NLS_LANG = AMERICAN_AMERICA.UTF8
   datafile location = /private1/oracle/oradata/NLDAP
   Dump destination =  /private1/oracle/app/oracle/admin/NLDAP/pfile,
                       /private1/oracle/app/oracle/admin/NLDAP/bdump,
                       /private1/oracle/app/oracle/admin/NLDAP/cdump,
                       /private1/oracle/app/oracle/admin/NLDAP/udump,
                       /private1/oracle/app/oracle/admin/NLDAP/create
```

Note: After installation of the Oracle database or Oracle directory, you use Oracle Database Configuration Assistant to create data file directories. Create the new directories on the new node under various UNIX partitions as defined by OFA.

Tasks To Be Performed on the Sponsor Node

Complete the following steps on the sponsor node.

1. At the command line prompt execute SQL*Plus.

```
$ sqlplus /nolog
SQL> connect /as sysdba
SQL> ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

The above command will create a trace file under the user dump destination directory (that is, `/private1/oracle/app/oracle/admin/LDAP/udump`).

The file will be created in the following format:

`$ORACLE_SID_`*`ora_processid`*`.trc`

For example:

`ldap_ora_4765.trc`

2. Shutdown the LDAP and replication servers and OID Monitor processes. Make sure the ldap and replication servers are stopped before stopping the OID Monitor process.

```
$ oidctl connect=net_service_name server=oidrepld instance=instance_number
stop
$ oidctl connect=net_service_name server=oidldapd instance=instance_number
stop
$ oidmon connect=net_service_name stop
```

In these commands, *net_service_name* is the net service name in the node's `tnsnames.ora` file.

3. On the remaining nodes, shutdown the LDAP replication server only.

```
$ oidctl connect=net_service_name server=oidrepld instance=instance_number
stop
```

Repeat the above procedure on all nodes except the sponsor node. Specify appropriate net service names for the corresponding nodes.

4. Quiesce **Oracle9i Replication** by running the following script at the **master definition site (MDS)**:

```
ldaprepl.sh -quiesce
```

When prompted, enter the Oracle global name and replication administration password for the MDS.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Note: This procedure can take place only on the Master Definition Site.

At this point, other nodes are available for LDAP edits only, but replication will not take place.

5. After quiescing the environment, shutdown the database and Oracle Net Services listener on the sponsor node only:

```
$ lsnrctl [listener_name] stop   (By default listener name is LISTENER)
$ sqlplus /nolog
SQL> connect /as sysdba
SQL> shutdown normal
SQL> exit
```

6. Copy the trace file created under Step 1 to a new file, `newdb.sql`, under the same directory.

```
$ cd $ORACLE_BASE/admin/LDAP/udump
$ cp ldap_ora_4765.trc newdb.sql
```

7. Edit `newdb.sql`, using any text editor, and delete the lines up to START NOMOUNT.

```
CREATE CONTROLFILE REUSE SET DATABASE database_name RESETLOG
```

8. Modify the UNIX directory location of the database/logfiles etc. to point to the new node directory. Refer to the sample file `newdb.sql` as follows:

```
Begin newdb.sql
CREATE CONTROLFILE REUSE SET DATABASE "LDAP" RESETLOGS
MAXLOGFILES 16
MAXLOGMEMBERS 2
MAXDATAFILES 255
MAXINSTANCES 1
MAXLOGHISTORY 100
LOGFILE
GROUP 1 '/private2/oracle/oradata/NLDAP1/log1_NLDAP.dbf'  SIZE 1M,
GROUP 2 '/private2/oracle/oradata/NLDAP1/log2_NLDAP.dbf'  SIZE 1M
DATAFILE
'/private2/oracle/oradata/NLDAP1/sys0_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/rbs1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/attrs1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/dncat1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/cncat1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/objcl1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/cats1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/default1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/temp1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/iattrs1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/idncat1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/icncat1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/iobjcl1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/icats1_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/temp2_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/cats2_NLDAP.dbf',
'/private2/oracle/oradata/NLDAP1/attrs2_NLDAP.dbf'
;
 End newdb.sql
```

9. Copy the files `initLDAP.ora` and `configLDAP.ora` under `$ORACLE_HOME/dbs` to `initNLDAP.ora` and `configNLDAP.ora` respectively.

```
$cd $ORACLE_HOME/dbs
$cp initLDAP.ora initNLDAP.ora
$cp configLDAP.ora configNLDAP.ora
```

10. Edit the copied file (`initNLDAP.ora`) and comment out the parameter JOB_QUEUE_PROCESS. Change the following parameter:

db_name = LDAP (If the parameter does not exist in the file initNLDAP.ora, then modify the file configNLDAP.ora)
ifile = *UNIX_directory_location_of_the_new_config_file/* configNLDAP.ora

11. Edit the copied file `configNLDAP.ora` to change the following parameters:

```
cdump =  UNIX_directory_location_of_the_new_node
udump  = UNIX_directory_location_of_the_new_node
bdump  = UNIX_directory_location_of_the_new_node
control_files = UNIX_directory_location_of_the_new_node
```

12. Edit the `tnsnames.ora` file to include information pertaining to the new node. Refer to the following sample file:

```
Begin tnsnames.ora

ldap1.world =
   (description=
      (address=(protocol=tcp)(host=rst-sun)(port=1521))
      (connect_data=(sid=LDAP))
   )
ldap2.world =
   (description=
      (address=(protocol=tcp)(host=eas-sun10)(port=1521))
      (connect_data=(sid=LDAP))
   )
ldap3.world =
   (description=
      (address=(protocol=tcp)(host=dsm-sun)(port=1521))
      (connect_data=(sid=NLDAP))
   )

End tnsnames.ora
```

13. Copy the file `listener.ora` to `list.bak`. Edit the copied file `list.bak` to include the information pertaining to the new node. Refer to the following sample file:

```
Begin listener.ora

# The KEY value for the IPC protocol may be anything, and
# is not related to either the TCP hostname or database SID.

LISTENER =
  (ADDRESS_LIST =
        (ADDRESS=(PROTOCOL= IPC)(KEY= LDAP))
```

```
        (ADDRESS=(PROTOCOL= IPC)(KEY= PNPKEY))
        (ADDRESS=(PROTOCOL= TCP)(Host= dsm-sun)(Port= 1521))
  )
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME= dsm-sun.us.oracle.com)
      (ORACLE_HOME= /private1/oracle/app/oracle/product/8.1.6)
      (SID_NAME = NLDAP)
    )
    (SID_DESC =
      (SID_NAME = extproc)
      (ORACLE_HOME = /private1/oracle/app/oracle/product/8.1.6)
      (PROGRAM = extproc)
    )
  )
STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = OFF

End listener.ora
```

The files `tnsnames.ora` and `listener.ora` can reside under `$ORACLE_HOME/network/admin` or `/var/opt/oracle` or under the directory pointed to by the TNS_ADMIN environment variable.

14. Copy the updated `tnsnames.ora` file to all the nodes. Be careful to copy it to the location of the current `tnsnames.ora` on each node. The file `tnsnames.ora` can be copied to other nodes using FTP. Make sure you transfer the file in ASCII mode.

Prior to copying the file `tnsnames.ora` to the new node, install the Oracle database software on the new node. Also copy the files `list.bak` as `listener.ora` and `sqlnet.ora` from the sponsor node to the new node.

15. Create an archive of all the data files and compress the archived file. For example:

```
$ >oradb.tar
```

This command will create an empty file under a directory. Make sure you have enough space in the partition where the archives will be created.

```
$ find / -name *.dbf -print -exec tar rvf  absolute_path_of_the_directory_
which_contains_oradb.tar {} \;
```

This command will search for all files ending with extension `.dbf` from the root directory. The assumption is that there is only one instance of the database server installed on the node and data files end with `*.dbf` extension.

```
$ find / -name *.log -print -exec tar rvf absolute_path_of_the_directory_
which_contains_oradb.tar
$ compress oradb.tar
```

This procedure is only an example to illustrate the method to back up the files. The Oracle data files will be backed up in the absolute path using this method. It is a better idea to back up the files from the current directory, so that you have more flexibility when you want to restore the data files. Consult your system administrator before backing up the database.

Tasks To Be Performed on the New Node

Complete the following steps on the new node.

1. Log in to the new node (dsm-sun).
2. Edit the `oratab` file appropriately for the new instance, at all database nodes. See the sample file for syntax.

   ```
   Begin oratab

   NLDAP:/private1/oracle/app/oracle/product/8.1.6:N
   *:/private1/oracle/app/oracle/product/8.1.6:N

   End oratab
   ```

3. Make sure the environment variables are set in the new directory site.
4. Install the Oracle database and Oracle directory server. Perform software only install of the Oracle database and directory server. Installation of Oracle database and directory software can be performed on the new node at any time before the database files are copied to the new machine. Perform post-installation (that is: `root.sh`) activities for the database as well as the Directory server.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

See Also: Oracle9*i* installation documentation

If you have already performed Oracle database and Directory installation on the new node, then proceed to Step 5.

5. Copy the files `initNLDAP.ora` and `configNLDAP.ora` from the sponsor node (rst-sun) to the new node under the UNIX directory `$ORACLE_BASE/ADMIN/NLDAP/PFILE`. Files can be copied to the new machine using tools such as FTP. Make sure the transfer mode is ASCII.

6. Create a symbolic soft link from `$ORACLE_HOME/DBS TO $ORACLE_BASE/ADMIN/NLDAP/PFILE`.

```
$ ln -s $ORACLE_BASE/admin/NLDAP/pfile/initNLDAP.ora
       $ORACLE_HOME/dbs/initNLDAP.ora
$ ln -s $ORACLE_BASE/admin/NLDAP/pfile/configNLDAP.ora
       $ORACLE_HOME/dbs/configNLDAP.ora
```

7. Copy the archived file created in the sponsor node procedure, using a tool such as FTP. (You created this file in Step 15 on page 24-7.) Set the transfer mode to binary.

```
ftp> open rst-sun
Connected to rst-sun.us.oracle.com.
220 rst-sun FTP server (UNIX(r) System V Release 4.0) ready.
Name (rst-sun:oracle):
331 Password required for oracle.
Password:
230 User oracle logged in.
ftp> cd /private1/oracle/oradata/LDAP
250 CWD command successful.
ftp> binary
200 Type set to I.
ftp> mget oradb.tar.Z
```

If the data files are huge (several gigabytes or terabytes) and the network bandwidth is low, then it may be a better idea to physically ship the compressed file on any media, such as tape or disk, from the sponsor to the new node.

8. Copy the file `newdb.sql` created under Step 6 of the sponsor node setup to the background user dump destination directory. You must transfer the file newdb.sql only in ASCII mode. For example:

```
 $ cd /private1/oracle/app/oracle/admin/NLDAP/udump
                    (that is::$ORACLE_BASE/admin/SID/udump)
$ ftp
ftp> open rst-sun
ftp> cd /private1/oracle/app/oracle/admin/LDAP/udump
ftp> mget newdb.sql
```

9. At the UNIX shell prompt execute the following commands:

```
$ sqlplus /nolog
SQL> connect /as sysdba
SQL> startup nomount
SQL> @newdb.sql
SQL> shutdown normal
SQL> startup (uncomment the parameter job_queue_process prior to  startup)
SQL>exit
$ lsnrctl start
```

10. Log in to the sponsor node and start up the database and listener on the sponsor node; for example, rst-sun.

```
$ telnet rst-sun
$ sqlplus /nolog
SQL> connect /as sysdba
SQL> startup
SQL> exit
$ lsnrctl start (By default listener name is LISTENER)
$ exit
```

11. If the sponsor node is a master site, then proceed to Step 12.

    If the new node is created by using backup database copy of the MDS, then the master definition catalog needs to be dropped and the underlying Oracle9*i* Replication catalogs must be created. To drop the definition of the MDS from the Oracle9*i* Replication catalog on the new node and add the Oracle9*i* Replication catalogs, execute the following scripts.

    ```
    $ cd $ORACLE_HOME/ldap/admin
    $ sqlplus repadmin/repadmin
    SQL> @ldapdropmds.sql
    SQL> @ldapcreindex.sql
    ```

    Specify the global name of the new node when prompted.

12. To configure the Oracle9*i* Replication, at the shell prompt, execute the following command:

    ```
    $ ldaprepl.sh -addnode
    ```

    Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

    - Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
    - MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

13. Update the LDAP replication agreements to include the new node.

    Sample LDIF file:

    ```
    dn: orclagreementid=000001, cn=orclreplagreements
    changetype: modify
    add: orcldirreplgroupdsas
    orcldirreplgroupdsas: dsm-sun
    ```

14. Start up the LDAP replication server on all the nodes, including new and sponsor nodes.

Verification Process

Log in to the Oracle database by using SQL*Plus and specify the user name as `ODS`, and the password `ods` when prompted.

Check the ods_chg_stat table on all nodes and see if they have correct and identical rows. The ods_chg_stat table should contain (*number of nodes*) x (*number of nodes*) rows. For example, if there were two nodes participating in Oracle9*i* Replication-based replication, and you added a third node, the ods_chg_stat table would contain nine rows, that is, 3 x 3, on each node. The rows are shown in the following table:

Supplier	Consumer	Change Number
Node1	node2	*number 1*
Node1	node3	*number 2*
Node1	node1	*number 3*
Node2	node1	*number 4*
Node2	node2	*number 5*
Node2	node2	*number 6*
Node3	node1	0
Node3	node2	0
Node3	node3	0

The rows with consumer names identical to that of suppliers contain the last changes processed by the outbound change log processing threads at the supplier sides. The rows with different supplier and consumer names contain last change numbers already processed from the suppliers to the consumers in question.

Since Node3 is a new node, there have been no changes supplied by Node3 yet. Therefore, the change numbers for Node3 as supplier are 0.

There may be a time delay before all nodes contain identical rows, but this delay should not be more than two to three minutes.

Part VI

The Directory and Clusters

This part contains these chapters:

25

Failover in Cluster Configurations

This chapter contains these topics:

Introduction

Oracle Internet Directory Release 9.0.2 enables you to increase high availability by using logical hosts—as opposed to physical hosts—in clustered environments.

A logical host consists of one or more disk groups, and pairs of host names and IP addresses. It is mapped to a physical host in the cluster. This physical host impersonates the host name and IP address of the logical host.

In this paradigm, the directory server binds to the logical host, rather than the physical host. It maintains this connection even if the logical host fails over to a new physical host.

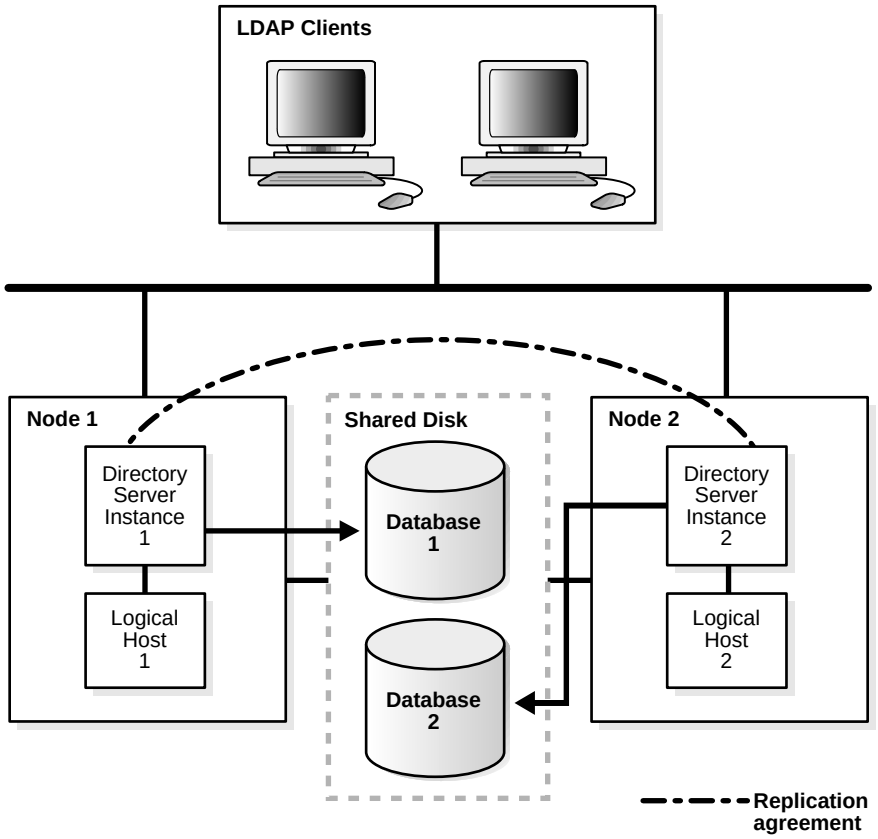
A client connects to the directory server by using the logical host name and address of the server. If the logical host fails over to a new physical host, then that failover is transparent to the client.

A logical host can reside on two or more cluster nodes that have physical access to its disk storage. A cluster can typically support any number of logical hosts, and a physical server or cluster node can impersonate more than one logical host.

This failover mechanism also supports replicated environments.

Figure 25–1 shows a sample Oracle Internet Directory configuration on a hardware cluster.

Figure 25–1 Oracle Internet Directory Configuration in a Two-Node Cluster



In this configuration:

- Physical Node 1 masters Logical Host 1
- Physical Node 2 masters Logical Host 2
- Directory Server Instance 1, consisting of one or more directory server instances, runs on Logical Host 1
- Directory Server Instance 2, consisting of one or more directory server instances, runs on Logical Host 2
- Both directory server instances have their respective directory data stores—Oracle databases—on the shared disk
- Directory Server Instance 1 and Directory Server Instance 2 are in a replication agreement

Clients connect to Directory Server Instance 1 by using the host name and address of Logical Host 1. Similarly, clients connect to Directory Server Instance 2 by using the host name and address of Logical Host 2.

Configuring Failover in a Clustered Environment

This section tells you how to configure failover in a clustered environment.

Note: At the end of Oracle Internet Directory installation, a directory server instance and the OID Monitor are started by default. To run Oracle Internet Directory on a logical host, you must stop the directory server instance and the OID Monitor, then restart them by using either of the optional flags `-host` or `-h`. Do this before any updates are made to the directory. This way, you ensure that the directory server uses the logical host name in change log generation.

It contains these topics:

- Step 1: Start OID Monitor
- Step 2: Start a Directory Server or Directory Replication Server by Using the OID Control Utility
- Step 3: Stop, then Restart, the Directory Server and OID Monitor

Step 1: Start OID Monitor

When you start OID Monitor, use the optional `host` argument, and set it to the logical host name. In the following example, OID Monitor connects to the directory store, `my_net_service` and monitors the directory server instances on the logical host, `my_host`.:

```
oidmon [connect=my_net_service] host=my_host
```

Step 2: Start a Directory Server or Directory Replication Server by Using the OID Control Utility

When you start the directory server by using the OID Control utility, use either of the optional flags `-host` or `-h`, and set it to the logical host name. In the following example, the OID Control utility directs the OID Monitor to start the directory server instance on the logical host, `my_host`.

```
oidctl connect=my_net_service server=oidldapd instance=1 flags="-h my_host"
start
```

Similarly, when you start a directory replication server by using the OID Control utility, use either of the optional flags `-host` or `-h`, and set it to the logical host name. In the following example, the OID Control utility directs the OID Monitor to start the directory replication server on the logical host, `my_host`.

```
oidctl connect=my_net_service server==oidrepld instance=1 flags="-h my_host"
start
```

Note: The replication agreement should use logical host names rather than physical host names.

Using logical hosts in a replicated environment requires a fresh Oracle Internet Directory installation. If you are upgrading from a replication environment earlier than release 3.0.1—in which host names in the replication agreement are different from the logical host names—then replication will not work.

Step 3: Stop, then Restart, the Directory Server and OID Monitor

To run Oracle Internet Directory on a logical host, stop the directory server instance and the OID Monitor, then restart them by using either of the optional flags `-host` or `-h`. Do this before any updates are made to the directory. This way, you ensure that the directory server uses the logical host name in change log generation.

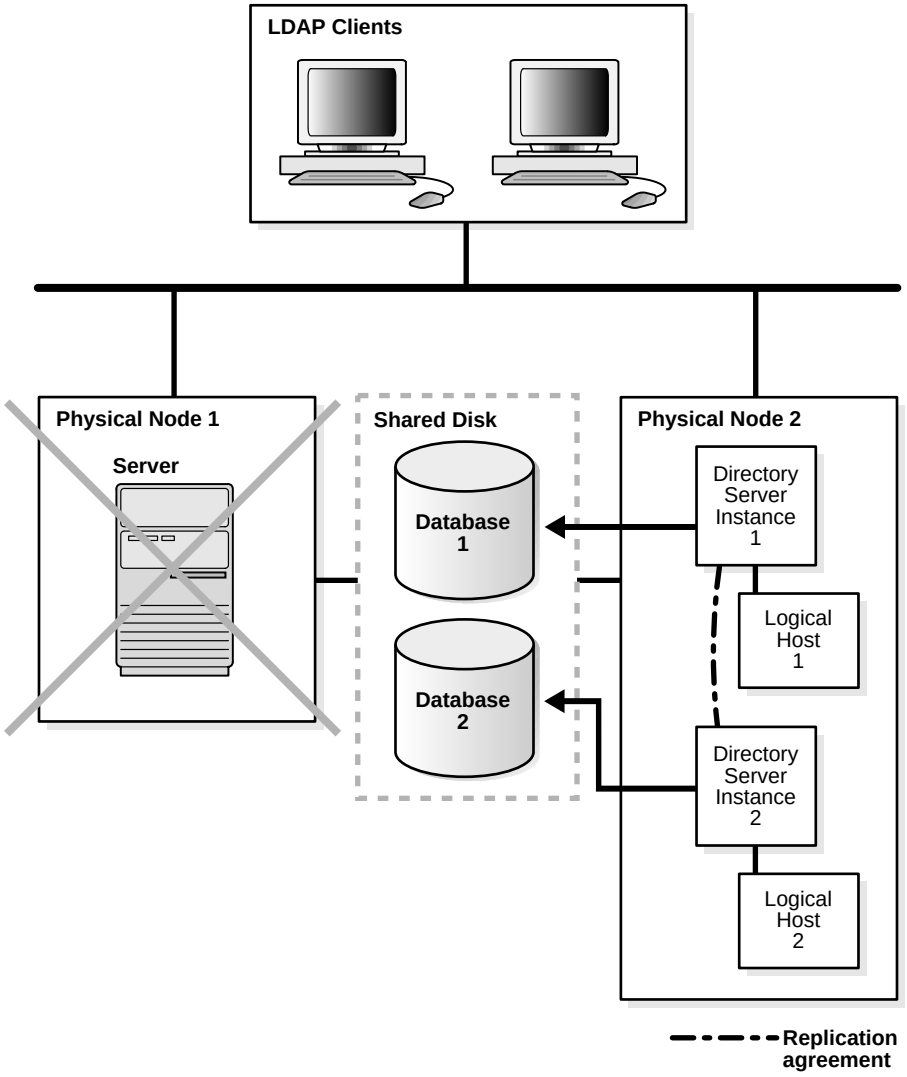
See Also:

- "Stopping an Oracle Directory Server Instance" on page 3-5
- "Stopping the OID Monitor" on page 3-3
- "Starting an Oracle Directory Server Instance" on page 3-4
- "Starting the OID Monitor" on page 3-2

How Failover Works in a Clustered Environment

Figure 25–2 shows a scenario in which a failover has occurred and the directory server has been restarted.

Figure 25–2 Oracle Internet Directory Nodes After Failover



In Figure 25–2, Physical Node 1 fails. At that point, Logical Host 1 fails over to be mastered by Physical Node 2. After this has finished, Directory Server Instance 1 needs to be restarted—that is, OID Monitor needs to be restarted with Logical Host 1 specified as the host name.

This failover of Directory Server Instance 1 is transparent to the LDAP clients connecting to Directory Server Instance 1. These clients continue to connect to Directory Server Instance 1 by using the host name and address of Logical Host 1.

After the failover, Directory Server Instance 1 continues to use the host name of Logical Host 1 in the change log generation. The replication agreement between Directory Server Instance 1 and Directory Server Instance 2 continues as before the failover.

26

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

Oracle9*i* Real Application Clusters is a computing environment that harnesses the processing power of multiple, interconnected computers. Along with a collection of hardware, called a cluster, it unites the processing power of each component to become a single, robust computing environment. A cluster comprises two or more computers, also called nodes.

This chapter discusses the ways you can run Oracle Internet Directory in an Oracle Real Application Clusters system. It contains these topics:

Terminology

- Node

  A computer where an instance resides. It can be part of a Massively Parallel Computing Infrastructure where it shares disk storage with other nodes. In most cases, a node has its own copy of the operating system.

- Cluster

  A set of instances, each typically running on different nodes, that coordinate with one another when accessing the shared database on the disk.

- Cluster Manager

  An operating system dependent component that discovers and tracks the membership state of nodes by providing a common view of cluster membership across the cluster.

- Transparent Application Failover (TAF)

  A runtime failover for high-availability environments, such as Oracle Real Application Clusters and Oracle Fail Safe, that refers to the failover and re-establishment of application-to-service connections. It allows client applications to automatically reconnect to the database if the connection fails, and optionally resume a SELECT statement that was in progress. This reconnect happens automatically from within the Oracle Call Interface (OCI)

  The client notices no connection loss as long as there is one instance left serving the application.

- Connect-time failover

  Failover method in which a client connect request is forwarded to a another listener if the first listener is not responding. It is enabled by service registration, because the listener knows if an instance is running before attempting a connection.

The Oracle Directory Server in an Oracle9*i* Real Application Clusters Environment

You can run a directory server on a node that is different from the one running the cluster database. The computer on which the directory server runs may be part of the cluster.

This section contains these topics:

- Oracle Internet Directory with Basic High Availability Configuration
- Oracle Internet Directory with Default N-Node Configuration

Oracle Internet Directory with Basic High Availability Configuration

In this case, a single directory server connects to two or more Real Application Clusters instances, each running on different nodes. This scenario is easy to configure and, on the node where the primary instance is running, it provides greater resilience after either a hardware or software failure.

Figure 26–1 shows the setup in detail.

Figure 26–1 Oracle Internet Directory with Basic High Availability Configuration

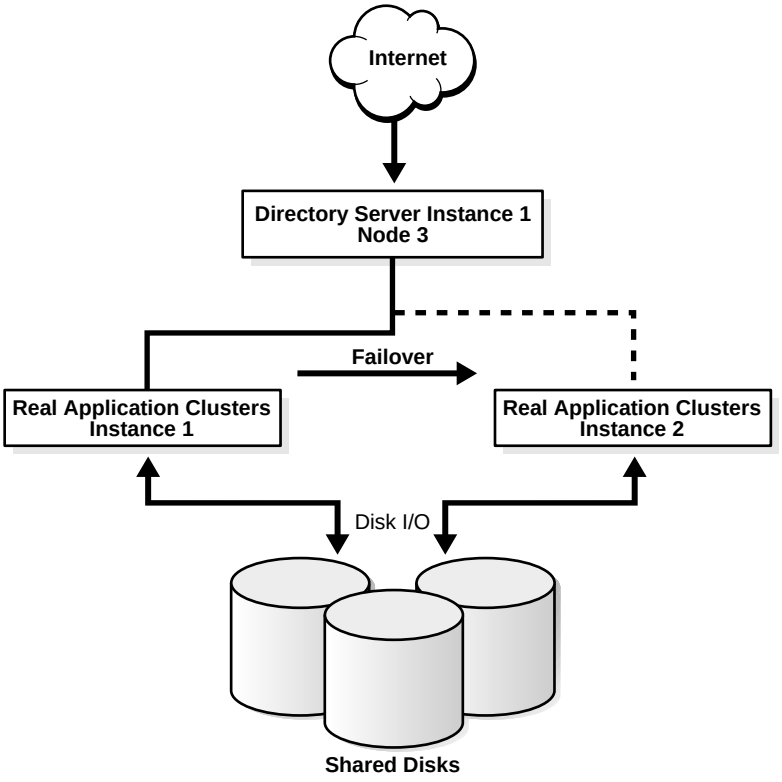


Figure 26–1 shows a three-node cluster. Real Application Clusters Instance 1 runs on Node 1. Real Application Clusters Instance 2 runs on Node 2. The directory server instance runs on Node 3.

Normally, the directory server instance communicates with the Real Application Clusters Instance on Node 1, which is the primary instance. However, in the event of either a hardware or software failure on a Node 1, Oracle Net Services can redirect database requests to the Real Application Clusters instance on Node 2, the secondary instance.

To specify the primary instance, in the initialization file, set the ACTIVE_INSTANCE_COUNT parameter to 1 for both instances. The instance you start first becomes the primary instance.

The primary instance can accept connections from its local listener, as well as from the secondary instance listener. A secondary instance registers with its local listener as a secondary instance, and like the primary instance, its ACTIVE_INSTANCE_COUNT parameter is set to 1. If the primary instance fails, then the secondary instance assumes the primary role and registers with its listeners. When the failed instance can once again start, it does so as the secondary instance. If you have failover configured, then directory server connections to the failed primary instance fail over to the secondary instance.

The following is an example of a tnsnames.ora file configured for a connect-time failover. In this example, the LOAD_BALANCE must be set to OFF.

```
MY_CLUSTER =
  (DESCRIPTION =
    (LOAD_BALANCE = OFF)
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = my_cluster.my_company.com))
  )
MY_CLUSTER_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_1)
    )
  )
MY_CLUSTER_2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_2)
    )
  )
```

The following is an example of a `listener.ora` file configured for a connect-time failover on my_host_1.

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
      )
    )
  )
```

The following is an example of a `tnsnames.ora` file configured for a transparent application failover (TAF).

```
MY_CLUSTER =
  (DESCRIPTION =
    (FAILOVER = ON)
    (LOAD_BALANCE = OFF)
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = my_cluster.my_company.com)
      (FAILOVER_MODE = (TYPE = SELECT) (METHOD = PRECONNECT)
                       (BACKUP = ops1))
    )
  )
MY_CLUSTER_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = ops1)
    )
  )
MY_CLUSTER_2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_2)
    )
  )
```

The following is an example of a `listener.ora` file configured for a transparent application failover (TAF) on my_host_1:

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
      )
    )
  )
```

Note: Depending on the state of the directory server when the database failure occurs, Oracle Internet Directory may not successfully manage the transparent application failover. In this case, Oracle Internet Directory logs "ORA-3113—End of file on Communication Channel" in the log file and re-establishes a new database connection against a live database instance.

If failover happens during any LDAP operation, then the client receives either the "DSA unwilling to perform" or the "ldapbind: Operations" error. When this happens, the client can simply reissue the request to the directory server.

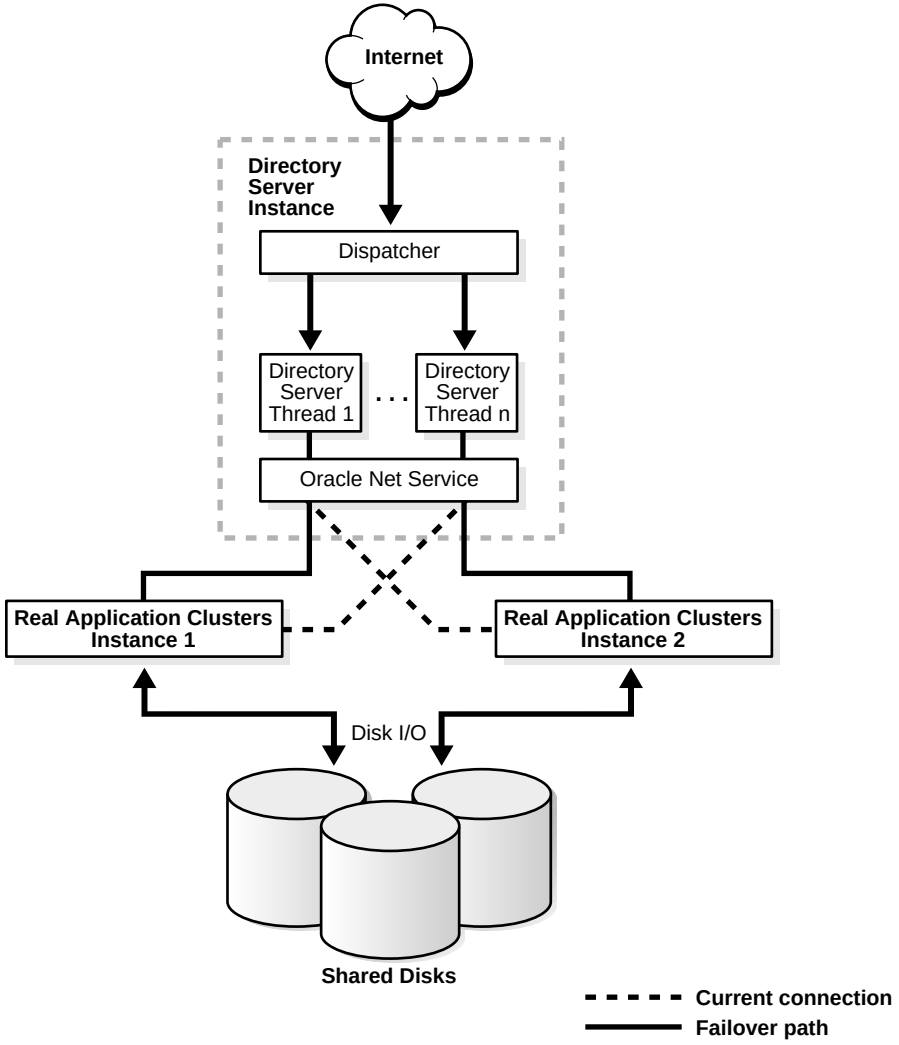
Oracle Internet Directory with Default N-Node Configuration

In this case, there are multiple directory server threads connecting to two or more Real Application Clusters instances on different nodes. To achieve this, you can set the `LOAD_BALANCE` parameter of Oracle Net Services to `ON`.

Figure 26–2 shows a three-node cluster. Real Application Clusters Instance 1 runs on Node 1. Real Application Clusters Instance 2 runs on Node 2. A directory server instance with Directory Server Thread #1 and Directory Server Thread #2 runs on Node #3.

See Also: *Oracle9i Net Services Administrator's Guide* in the Oracle Database Documentation Library for instructions on setting the `LOAD_BALANCE` parameter

Figure 26–2 Single Directory Server Instance on One Node and Multiple Real Application Clusters Instances



Depending on how the Oracle Net Services routes the LDAP request, when all nodes in Figure 26–2 are running, directory server thread 1 may connect to Real Application Clusters Instance 1, and directory server thread 2 may connect to Real Application Clusters Instance 2. Incoming LDAP requests to the directory server are distributed in round-robin fashion to both directory database connections. If there is a hardware or software failure on node 1, then directory server thread 1 reconnects to Real Application Clusters Instance 2 by using connection time failover or Oracle Net transparent application failover.

The scenario in this example provides higher availability and scalability. If the database or database host fails, then it provides resilience by using connection time failover or Oracle Net transparent application failover. Moreover, it provides higher throughput for complicated LDAP subtree searches.

To configure your system for this scenario, examine the following examples of the various configuration files.

The following example shows a `tnsnames.ora` file configured for a connect-time failover.

```
MY_CLUSTER =
  (DESCRIPTION =
    (LOAD_BALANCE = ON)
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = my_cluster.my_company.com))
  )
MY_CLUSTER_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_1)
    )
  )
MY_CLUSTER_2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_2)
    )
  )
```

The following example shows a `listener.ora` file configured for connect time failover on my_host_1:

```
# LISTENER.ORA Network Configuration
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
      )
    )
  )
```

The following two examples show two `tnsnames.ora` files, one for `my_host_1` and the other for `my_host_2`, configured for a transparent application failover (TAF).

The `tnsnames.ora` on `my_host_1`:

```
MY_CLUSTER =
  (DESCRIPTION =
    (FAILOVER = ON)
    (LOAD_BALANCE = ON)
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = my_cluster.my_company.com)
      (FAILOVER_MODE = (TYPE = SELECT) (METHOD = PRECONNECT)
                       (BACKUP = my_host_1))
    )
  )
```

```
MY_CLUSTER_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_company_1)
    )
  )
MY_CLUSTER_2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_company_2)
    )
  )
```

The `tnsnames.ora` on `my_host_2`:

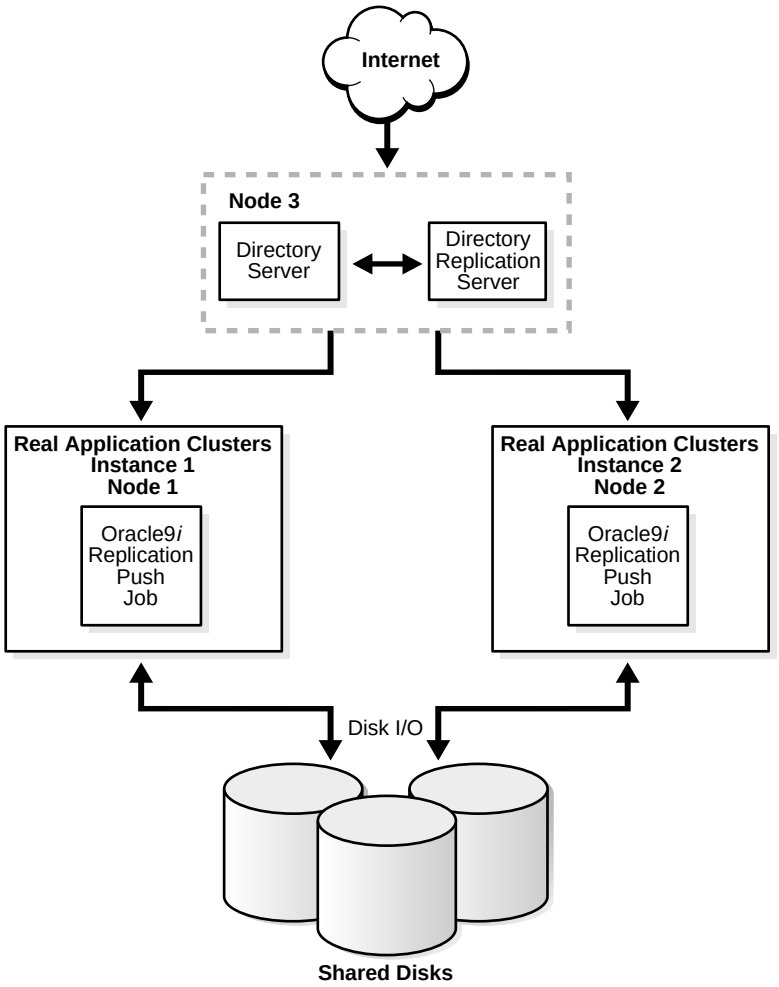
```
MY_CLUSTER =
  (DESCRIPTION =
    (FAILOVER = ON)
    (LOAD_BALANCE = ON)
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = my_cluster.my_company.com)
      (FAILOVER_MODE = (TYPE = SELECT) (METHOD = PRECONNECT)
                       (BACKUP = my_company_2))
    )
  )
MY_CLUSTER_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_1)
    )
  )
```

```
MY_CLUSTER_2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = my_host_2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = my_cluster.my_company.com)
      (INSTANCE_NAME = my_cluster_2)
    )
  )
```

The Oracle Directory Replication Server in an Oracle9*i* Real Application Clusters Environment

Figure 26–3 shows a possible Oracle directory replication server in an Oracle Real Application Clusters environment.

Figure 26–3 Directory Replication Server in an Oracle Real Application Clusters Environment



There are three nodes in this configuration. A directory server instance runs on Node 3, and Real Application Clusters instances run on Node 1 and Node 2. When all nodes are running, the directory replication server connects to the directory server instance, and the Oracle9*i* Replication push jobs are running on both Real Application Clusters instances. If there is any hardware failure on Node 3, the directory replication server on Node 2 restarts and connects to directory server instance 2. If any hardware failure happens on Node 2, then, after cluster reconfiguration, the Oracle9*i* Replication push job continues on Real Application Clusters instance 1.

This scenario provides resilience in the event of database or database host failure for replication data transfer, that is, an Oracle9*i* Replication push job. It also provides resilience in the event of a directory server instance or host failure, or failure for the directory replication server.

Part VII

Directory Plug-ins

This part contains this chapter:

27

Oracle Internet Directory Plug-in Framework

This chapter describes how you can extend the capabilities of the Oracle directory server by using plug-ins developed by either Oracle Corporation or third-party vendors.

This chapter contains these topics:

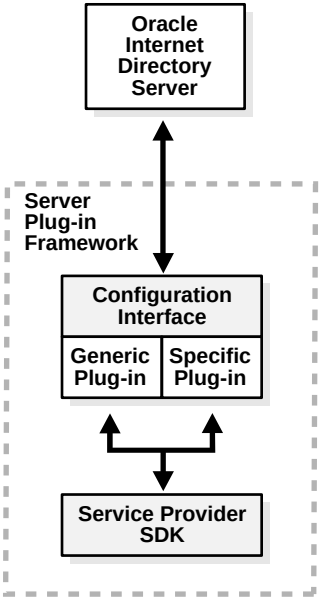
About Directory Server Plug-ins

Oracle Internet Directory supports a directory server plug-in a PL/SQL package. It can add the following kinds of functionality to the directory server, to mention just a few:

- Validate data before the directory server performs an operation on it
- Perform specified actions after the server performs an operation
- Define password policies
- Authenticate users through external credential stores

On startup, the directory server loads your plug-in configuration and library. Then, when it processes requests, it calls your plug-in functions whenever the specified event takes place.

Figure 27–1 Oracle Internet Directory Server Plug-in Framework



Operation-Based Plug-ins

This section describes the operation-based plug-ins that the Oracle Internet Directory plug-in framework supports. These plug-ins execute before, after, or in addition to normal directory server operations.

Table 27–1 Types of Operation-Based Plug-ins

Type of Plug-in	Description
Pre-operation	Plug-ins that the directory server calls *before* performing an LDAP operation. Typically, these plug-ins validate data before using it in an LDAP operation. If validation fails, then depending on the error or warning returned from the plug-in, the LDAP operation can decide to proceed or not. However, if the associated LDAP request fails later on, then Oracle Internet Directory does not roll back whatever the plug-in has already committed.
Post-operation	Plug-ins that the directory server calls *after* performing an LDAP operation. Typically, these plug-ins invoke a function, such as logging or notification, when the directory server performs a particular operation. If the plug-in fails, then the directory server does not roll back the associated LDAP operation. The plug-in executes regardless of whether the associated LDAP request fails.
When-operation	Plug-ins that the directory server calls in addition to standard processing. Typically, these plug-ins augment existing functionality, performing extra operations in the same transactions as the corresponding LDAP operations. If either the LDAP operation or the plug-in fails, then the directory server rolls back the changes.

Registering Plug-ins

To enable the directory server to call a plug-in at the right moment, you must register the plug-in with the directory server. Do this by creating an entry for the plug-in under `cn=plugin,cn=subconfigsubentry`.

The orclPluginConfig Object Class

A plug-in must have `orclPluginConfig` as one of its object classes. This is a structural object class, and its super class is `top`. Table 27–2 lists and describes its attributes.

Table 27–2 Attributes of the orclPluginConfig Object Class

Attribute	Attribute Value	Mandatory	Optional
`Cn`	Plug-in entry name	X	
`orclPluginName`	Plug-in package name	X	

`orclPluginType`	One of the following values:
	`operational`
	`attribute`
	`password_policy`
	`syntax`
	`matchingrule`
	See Also: Operation-Based Plug-ins on page 27-3
		X	

`orclPluginKind`	PL/SQL		X
`orclPluginEnable`	0 = disable (default)		
	1 = enable		X
`orclPluginVersion`	Supported plug-in version number		X
`orclPluginShareLibLocation`	File location of the dynamic linking library. If this value is not present, then the directory server assumes that the plug-in language is PL/SQL.		X

`orclPluginLDAPOperation`	One of the following values:
	`ldapcompare`
	`ldapmodify`
	`ldapbind`
	`ldapadd`
	`ldapdelete`
	`ldapsearch`
			X

Table 27–2 Attributes of the orclPluginConfig Object Class

Attribute	Attribute Value	Mandatory	Optional
orclPluginTiming	One of the following values: pre when post		X
orclPluginIsReplaced	0 = disable (default) 1 = enable For WHEN timing plug-in only		X
orclPluginSubscriberDNList	A semicolon separated DN list that controls whether the plug-in takes effect. If the target DN of an LDAP operation is included in the list, then the plug-in takes effect.		X

Adding a Plug-in Entry Using Command-Line Tools

Plug-ins must be added to the Oracle Internet Directory server so that the server is aware of additional operations that must be performed at the correct time.

When the plug-in successfully compiles against the Oracle Internet Directory back end database, create a new entry and place it under `cn=plugin,cn=subconfigsubentry`.

In the following examples, an entry is created for an operation-based plug-in called `my_plugin1`. The LDIF file, `my_ldif_file.ldif`, is as follows:

Example 1

The following is an example LDIF file to create such an object:

```
cn=when_comp,cn=plugin,cn=subconfigsubentry
objectclass=orclPluginConfig
objectclass=top
orclPluginName=my_plugin1
orclPluginType=operational
orclPluginTiming=when
orclPluginLDAPOperation=ldapcompare
orclPluginEnable=1
orclPluginVersion=1.0.1
orclPluginIsReplace=1
cn=when_comp
```

```
orclPluginKind=PLSQL
orclPluginSubscriberDNList=dc=COM,c=us;dc=us,dc=oracle,dc=com;dc=org,dc=us;o=IMC
,c=US
```

Example 2

```
cn=post_mod_plugin, cn=plugin,cn=subconfigsubentry
objectclass=orclPluginConfig
objectclass=top
orclPluginName=my_plugin1
orclPluginType=operational
orclPluginTiming=post
orclPluginLDAPOperation=ldapmodify
orclPluginEnable=1
orclPluginVersion=1.0.1
cn=post_mod_plugin
orclPluginKind=PLSQL
```

Add this file to the directory with the following command:

```
ldapadd -p 389 -h myhost -D binddn -w password -f my_ldif_file.ldif
```

When you have added this entry to the directory, the directory server validates the plug-in by quickly executing it and checking for compilation or access privilege errors. It then gathers more information about this plug-in—such as attributes, timing, or the type of LDAP operation related to the plug-in.

Notes: The plug-in configuration entry, for example, `cn=plugin,cn=subconfigsubentry` metadata is not replicated in the replication environment to avoid creating inconsistent state.

See Also: Oracle Internet Directory *Application Developer's Guide* for more detail

Part VIII

The Oracle Directory Integration Platform

This part explains the concepts, architecture, and components of the Oracle Directory Integration Platform, and tells you how to configure and use it to synchronize multiple directories with Oracle Internet Directory. It contains these chapters:

28

Oracle Directory Integration Platform Concepts and Components

This chapter introduces the Oracle Directory Integration Platform: its components, structure, and administration tools.

This chapter contains these topics:

What Is the Oracle Directory Integration Platform?

The Oracle Directory Integration Platform enables an enterprise to integrate its applications and other directories with Oracle Internet Directory. This platform provides all the interfaces and infrastructure necessary to keep the data in Oracle Internet Directory consistent with the data in enterprise applications and connected directories.

For example, an enterprise might want employee records in its Oracle Human Resources database to be synchronized with Oracle Internet Directory. In addition, the enterprise may deploy certain LDAP-enabled applications (such as Oracle9*i*AS Portal) that need to be notified whenever changes are applied to Oracle Internet Directory. This service is called provisioning, and the Oracle Directory Integration Platform provides such applications with the necessary notifications.

Based on the nature of integration, the Oracle Directory Integration Platform provides two distinct services:

- The synchronization integration service, which keeps connected directories consistent with the central Oracle Internet Directory
- The provisioning integration service, which sends notifications to target applications periodically to reflect changes made to a user's status or information

These services are described and illustrated in later sections.

Why is the Oracle Directory Integration Platform Needed?

Using Oracle Internet Directory as the central repository for diverse LDAP-enabled applications and connected directories can greatly reduce your time and resource costs for administration. To realize these benefits, however, requires the services described above, which ensure that these connected entities reliably receive (and provide) the necessary information. The following scenarios, two for synchronization and two for provisioning, illustrate how these needs may arise and be met:

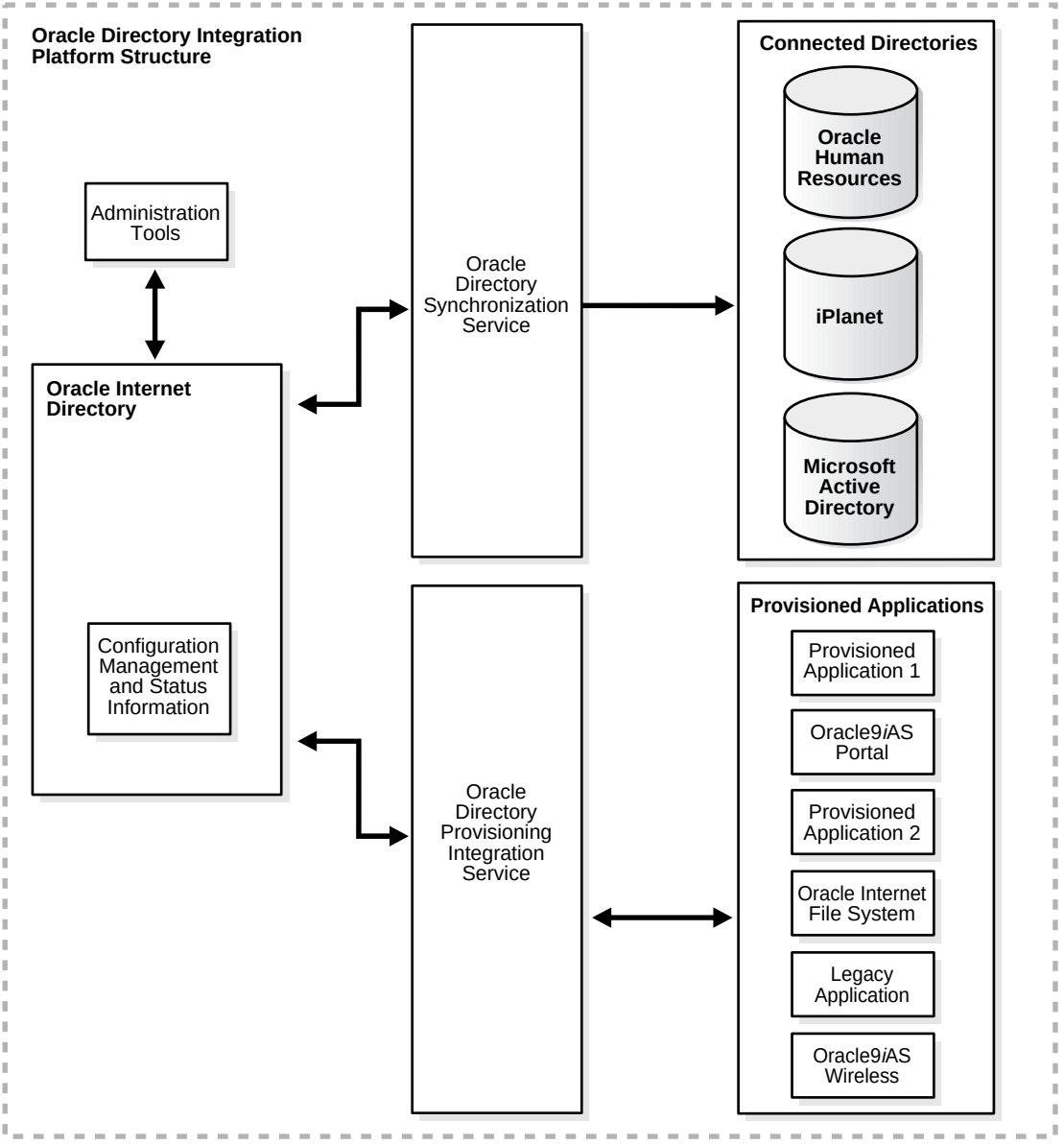
- Synchronization: An enterprise may wish to deploy LDAP-enabled applications from Oracle in the presence of a third-party LDAP directory. But Oracle applications are only certified to run against Oracle Internet Directory. The deployment will thus need to synchronize data between Oracle Internet Directory and the third-party directory.

- Synchronization: An enterprise may wish to deploy a metadirectory solution that integrates several repositories in the enterprise with Oracle Internet Directory, thus requiring synchronization.
- Provisioning: An enterprise may wish to deploy certain LDAP-enabled Oracle components such as Oracle9*i*AS Portal, Oracle Internet File System, and Oracle9*i*AS Wireless. The user/group provisioning for these components is integrated with Oracle Internet Directory, requiring they be notified of user or group changes in that repository.
- Provisioning: An enterprise may be developing and deploying custom relational applications that need to be notified of changes to user/group entries in the central Oracle Internet Directory. This need is met by the provisioning integration service of the Oracle Directory Integration Platform.

Structure of the Oracle Directory Integration Platform

Figure 28–1 shows the structure of the Oracle Directory Integration Platform.

Figure 28–1 Oracle Directory Integration Platform Structure



The sections that follow describe each component and its relation to the rest of the Oracle Directory Integration Platform.

Provisioning versus Synchronization

Provisioning notifies applications of changes to user or group entries or attributes that the application needs to track. Synchronization deals with directories rather than applications, ensuring the consistency of entries and attributes that reside in both Oracle Internet Directory and other connected directories.

This section contains these topics:

Provisioning

Provisioning is the service you need when you are designing or installing an application that

- Does not maintain a directory
- Is LDAP-enabled
- Can and should allow only authorized users to access its resources

The goal of provisioning is to ensure that the application is notified of changes to user or group information. Such changes can affect whether the application allows a user access to its processes and which resources can be used.

A provisioning integration profile must be created during the application's installation. The Provisioning Subscription Tool enables you to specify the necessary information and then creates that profile.

Synchronization

You choose synchronization to coordinate changes among Oracle Internet Directory and connected directories. The goal of synchronization is to share and make consistent any change to directory information, including data elements other than a user's name, group memberships, or privileges. For all directories to both use and provide only the latest data, every directory must be informed of each such change made in any connected directory.

Whenever you decide to connect a directory to Oracle Internet Directory, a synchronization profile must then be created for that specific directory. It specifies the format and content of the notifications exchanged between Oracle Internet Directory and the directory to be connected.

How Provisioning and Synchronization Differ

Provisioning and synchronization have important operational differences. Critical actions must be taken at different times. Different maintenance effort levels are required. Communication differs in being one-way or two-way, and the types of data to be handled are different. Table 28–1 provides a brief tabular format for these primary distinctions.

Table 28–1 Provisioning Integration and Directory Synchronization Distinctions

Service	Provisioning Integration	Directory Synchronization
The time for action	Application design time. Provisioning Integration is targeted towards application designers who are developing LDAP-enabled applications.	Application deployment time. Directory synchronization is targeted towards connected directories that need to be synchronized with Oracle Internet Directory.
Maintenance effort	Minimal: need only register the application end-point during install	High: need to set up the mapping rules and the agents
Communication direction	One way, from Oracle Internet Directory to provisioned applications	Two-way: from Oracle Internet Directory to connected directories and / or vice versa
Type of data	Restricted to provisioned Users and Groups	Any data in a directory
Example	Oracle9*i*AS Portal	Oracle Human Resources

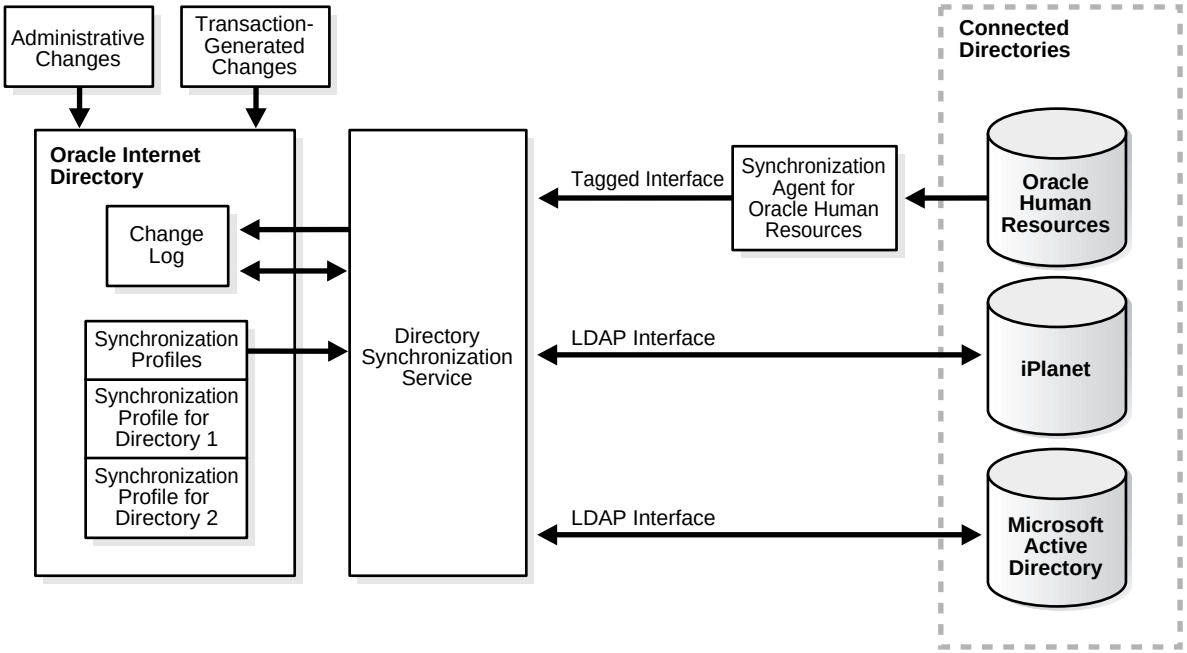
Directory Synchronization Service

In the Oracle Directory Integration Platform environment, connected directories are those whose contents are synchronized with Oracle Internet Directory through the Oracle Directory Synchronization Service.

Oracle Internet Directory is the central directory for all information, with which all other directories are synchronized. This synchronization can be bidirectional: changes in Oracle Internet Directory can be exported to connected directories, and changes in connected directories can be imported into Oracle Internet Directory. However, some connected directories (such as Oracle Human Resources) do not

receive changes from Oracle Internet Directory, though they supply changes to Oracle Internet Directory. Selective attributes can be targeted (or ignored) by the synchronization service. For example, employee badge numbers appear in Oracle Human Resources but have no relevance to Oracle Internet Directory or its connected directories or client applications. On the other hand, employee id number does have relevance or utility, and does get synchronized by the service.

Figure 28–2 Interactions of the Directory Synchronization Service



The central mechanism triggering all such synchronization activities is the Change Log. Every change to any connected directory, including Oracle Internet Directory, is reflected by one or more entries in the Change Log. The Directory Synchronization Service checks the Change Log periodically, taking action whenever a change corresponds to one or more Synchronization Profiles. The service then supplies the appropriate change to all other connected directories whose individual Profiles correspond to the logged change.

Such directories could include, for example, relational databases, Oracle Human Resources, Microsoft Exchange, or Lotus Notes. Synchronization through Oracle Directory Integration connectors ensures that Oracle Internet Directory remains

up-to-date with all information that Oracle Internet Directory clients need it to have.

Provisioning Integration Service

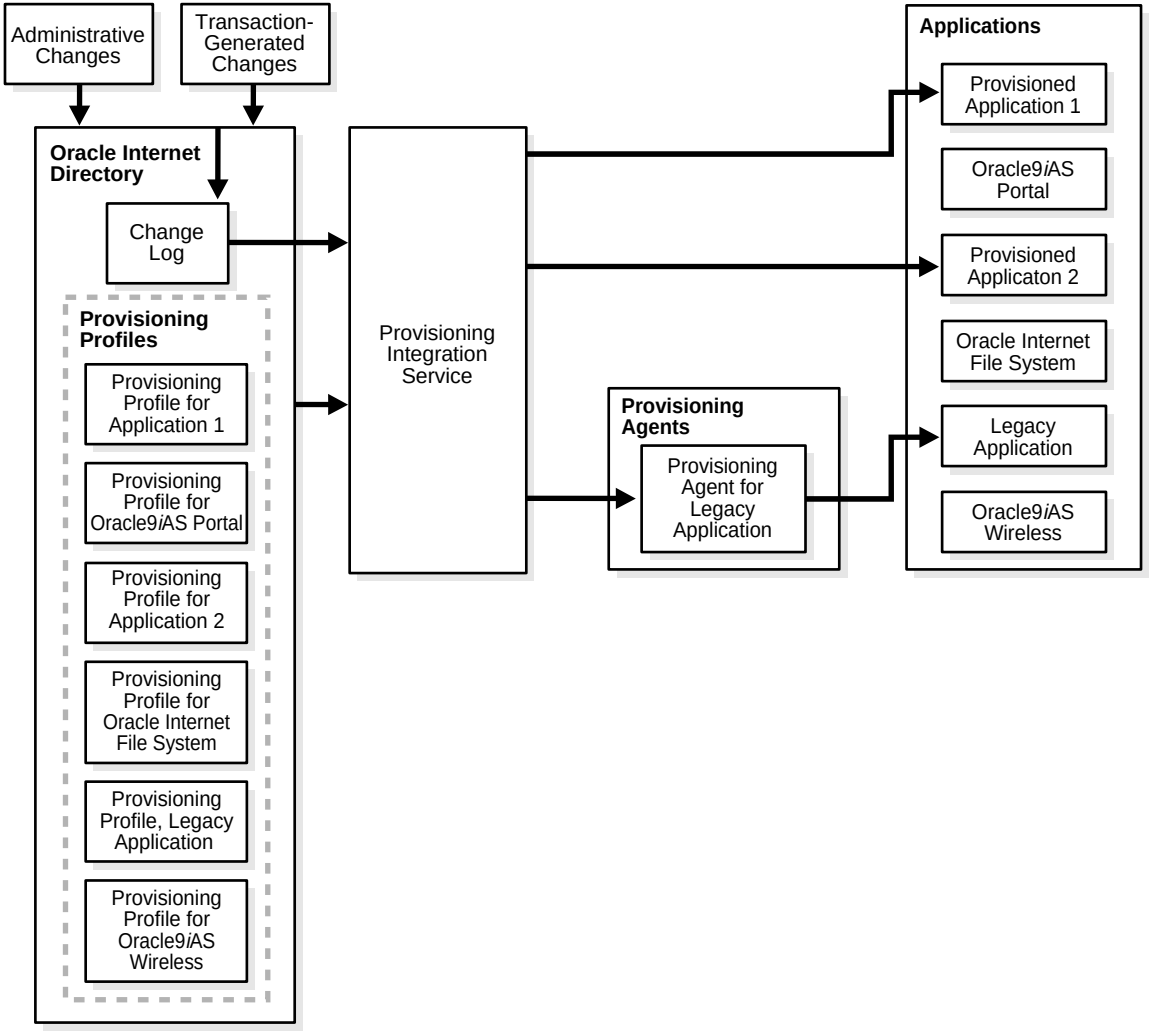
The provisioning integration service requires a provisioning profile for each application that is to be notified of changes in user or group information. Each provisioning profile uniquely identifies the application and organization to which it applies, and specifies the users, groups, and operations requiring the application to be notified. The profile must be created when the application is installed, using the Subscription Tool described in Appendix A, "Syntax for LDIF and Command-Line Tools".

When changes are made in Oracle Internet Directory that match an application's provisioning profile, the provisioning integration service sends the relevant data to that application, such as Oracle9*i*AS Portal.

The term "legacy application" means one already operational before this Service was installed, and therefore one that has not subscribed in the usual way, during installation. To enable such an application to receive provisioning information by means of the provisioning integration service, a provisioning agent must be developed in addition to the provisioning profile. The agent must be specifically designed and built to translate the relevant data taken from Oracle Internet Directory into the exact format required by the legacy application.

Figure 28–3 illustrates these interactions, including the special case of a provisioning agent used for a legacy application.

Figure 28–3 Interactions of the Provisioning Integration Service



See Also: Chapter 36, "The Oracle Directory Provisioning Integration Service" for more details about the Oracle Directory Provisioning Integration Service

Oracle Directory Integration Server

The Oracle Directory Integration Server is the multithreaded server process consisting of the two services described earlier: the Oracle Directory Synchronization Service and the Oracle Provisioning Integration Service.

The Oracle Directory Integration Server performs the following functions for the Oracle Directory Synchronization Service:

- Scheduling—Processing a Synchronization Profile based on a predefined schedule
- Mapping—Executing rules for converting data between connected directories and Oracle Internet Directory
- Data propagation—Sending data to connected directories using a connector
- Error handling

Oracle Directory Integration Server performs the following functions for the Oracle Directory Provisioning Integration Service:

- Scheduling—Processing a Provisioning Profile based on a predefined schedule
- Event Notification—Notifying an application of a relevant change to the user or group data stored in Oracle Internet Directory
- Error handling

> **See Also:** Chapter 30, "Oracle Directory Integration Server Administration"

Directory Integration Toolkit

The directory integration toolkit allows third party vendors and developers to integrate their solutions with the Oracle Directory Integration Platform environment. Such vendors can include providers of metadirectories and provisioning solutions. The toolkit also allows application vendors whose products are based on or use the Oracle technology to integrate provisioning of their users and groups with Oracle Internet Directory.

The toolkit describes the following interfaces, tools, and procedures:

- Interfaces for accessing changes in Oracle Internet Directory by clients:
  - IETF standard change log interface
  - Oracle proprietary change log interface

- Interfaces to register or modify directory integration connectors in Oracle Internet Directory, for scheduling or data mapping, using either:
  - Oracle Directory Manager, or
  - Command-line tools to add and modify data by using an LDIF file configuration
- Tools and procedures for bootstrapping connected directories into the Oracle Directory Integration Platform environment. These enable you to:
  - Bulk import data from LDIF files
  - Bulk export Oracle Internet Directory data into LDIF files
- Interfaces to subscribe to user / group Provisioning Events (changes) in Oracle Internet Directory
- Interfaces to consume events being sent by the provisioning integration service

Administration and Monitoring Tools

This section contains these topics:

- Oracle Directory Manager
- OID Control and OID Monitor
- Oracle Enterprise Manager

Oracle Directory Manager

Oracle Directory Manager, a Java-based graphical user interface tool, enables you to administer the Oracle Directory Integration Platform. Specifically, it enables you to:

- Create, modify, and delete directory integration profiles
- Check the status of directory integration profiles
- Check the status of all Oracle directory integration server instances

See Also: Chapter 4, "Directory Administration Tools"

OID Control and OID Monitor

OID Control and OID Monitor enable you to start, stop, and monitor Oracle directory integration server.

In Oracle Internet Directory, you can use OID Control and OID Monitor to control the directory integration server in the ORACLE_HOME where either the Oracle directory server or Oracle directory integration server are installed. If Oracle Internet Directory installation is client-only, then the OID Control utility and OID Monitor are not installed. In this case, start Oracle directory integration server manually. In this configuration you can still use Oracle Directory Manager to learn the status of Oracle directory integration server.

See Also:

- Chapter 3, "Preliminary Tasks and Information"
- Chapter 4, "Directory Administration Tools"
- Chapter 30, "Oracle Directory Integration Server Administration"

Oracle Enterprise Manager

You can use Oracle Enterprise Manager to monitor the status of various integration profiles. This integrated, comprehensive systems management platform combines a graphical console, agents, common services, and tools for scheduling, monitoring, and administering your heterogeneous environment.

See Also: the following for more details:

- *Oracle Enterprise Manager Concepts Guide*
- *Oracle Enterprise Manager Administrator's Guide*
- Oracle Enterprise Manager online help

Sample Deployment of the Oracle Directory Integration Platform

This section describes a deployment in which the Oracle Directory Integration Platform is used for integrating various applications in the enterprise. This enterprise has the following components:

- Oracle Human Resources system, in which all employees and contractors are managed.
- An existing iPlanet Directory Server, which is being used by certain applications.
- An installation of Oracle9*i*AS Portal, which is used as the intranet portal for all employees.
- An installation of Oracle Internet File System Release 9.0.2, which is used as a document repository for all corporate documents.

The enterprise has the following functional requirements:

1. A single source of truth for all employee records. The deployment would like all employees and contractors to be created in Oracle Human Resources. Once created, the deployment would like all applications in the enterprise to share this information through Oracle Internet Directory.
2. When a user gets created in Oracle Human Resources, all applications in the enterprise including single sign-on services should be able to honor the employee.
3. When changes to user properties are made, all applications that are interested in such changes should be notified.

4. When a user gets terminated in Oracle Human Resources, the deployment would like all access rights of the user to be revoked.

Overall Deployment

Figure 28–4 illustrates the various components and their relationships to each other:

Figure 28–4 Example of Oracle Directory Integration Platform in Deployment

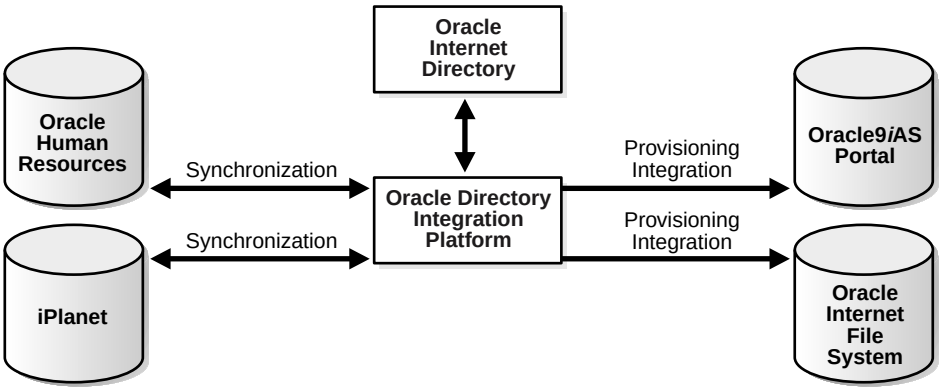


Figure 28–4 illustrates the following factors:

1. Oracle Internet Directory is used as the centralized user repository for all enterprise applications.
2. Oracle Human Resources is the source of truth for all user related information. It is being synchronized with Oracle Internet Directory using the Directory synchronization service of the Oracle Directory Integration Platform.
3. iPlanet Directory Server, which is already deployed in the enterprise is now being synchronized with Oracle Internet Directory using the directory synchronization service of the Oracle Directory Integration Platform.
4. Oracle9*i*AS Portal is being notified of changes in Oracle Internet Directory by using the Provisioning Integration Service of the Oracle Directory Integration Platform.
5. Oracle Internet File System is also being notified of changes in Oracle Internet Directory using the Provisioning Integration Service of the Oracle Directory Integration Platform.

The sections that follow describe the flow of information during user creation, modification, and deletion, thereby illustrating the various capabilities of the Oracle Directory Integration Platform.

User Creation and Provisioning

Based on the requirements specified by the deployment, all users are created in Oracle Human Resources. It is the responsibility of the Oracle Directory Integration Platform to propagate new user records to all other repositories in the enterprise. Figure 28–5 illustrates the various interactions that help the Oracle Directory Integration Platform complete this task:

Figure 28–5 User Creation and Provisioning

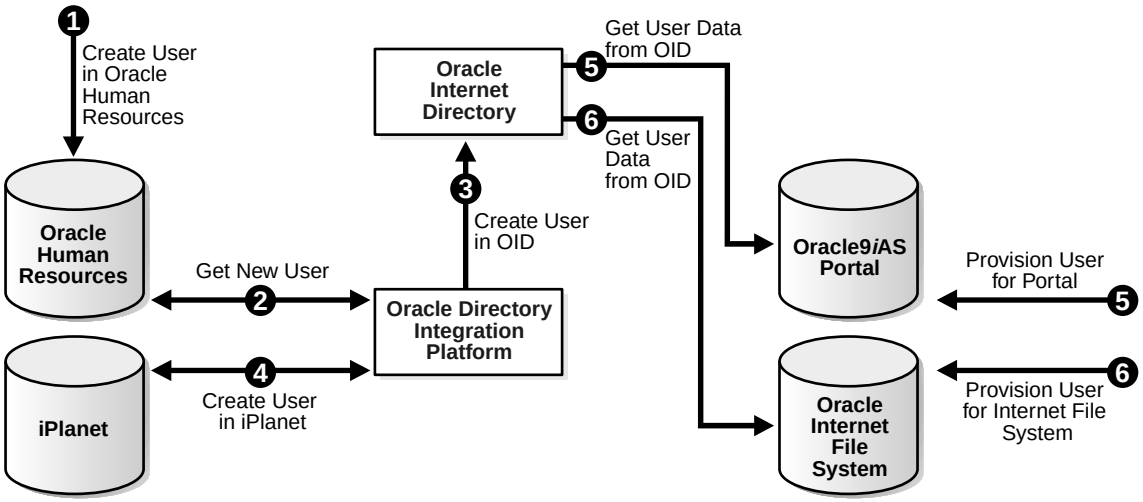


Figure 28–5 shows the creation of a new user in Oracle Human Resources, which causes an entry for that user to be created in Oracle Internet Directory and the iPlanet directory servers. It also shows the process of provisioning the user to access Oracle9*i*AS Portal and Oracle Internet File System deployed in the enterprise. User creation and provisioning happen in the following manner:

1. First the Oracle Human Resources administrator of the company creates the user in the Oracle Human Resources database.
2. The synchronization integration service of the Oracle Directory Integration Platform detects the new user creation.

3. The synchronization integration service then creates the entry for the user in Oracle Internet Directory.
4. The synchronization integration service also creates an entry in the iPlanet directory.
5. Since the user entry is available in Oracle Internet Directory, the Oracle9*i*AS Portal administrator can now provision the user to use the services of Oracle9*i*AS Portal. During this task, the Oracle9*i*AS Portal software automatically fetches the user details from Oracle Internet Directory.
6. The Oracle Internet File System administrator also provisions the user to use Oracle Internet File System services by using a similar process.

Note that the Oracle Directory Integration Platform does not directly notify Oracle9*i*AS Portal or Oracle Internet File System about new users. This is because not all users created in Oracle Human Resources need access to all services. In this case, the deployment must explicitly provision the users to use these services, as in steps 5 & 6.

Modification of User Properties

Based on the requirements of the deployment, any modification to user properties must be communicated to all components interested in such changes. Figure 28–6 illustrates the actions the Oracle Directory Integration Platform takes to meet this requirement.

Figure 28–6 Modification of User Properties

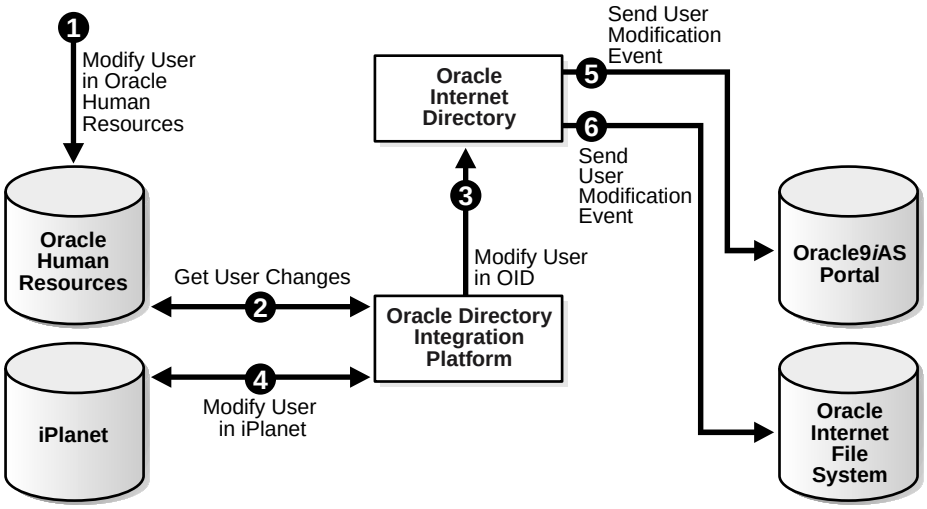


Figure 28–6 shows the process by which Oracle Directory Integration Platform communicates the modification of user properties to all systems in the enterprise. The process contains the following sequence of events:

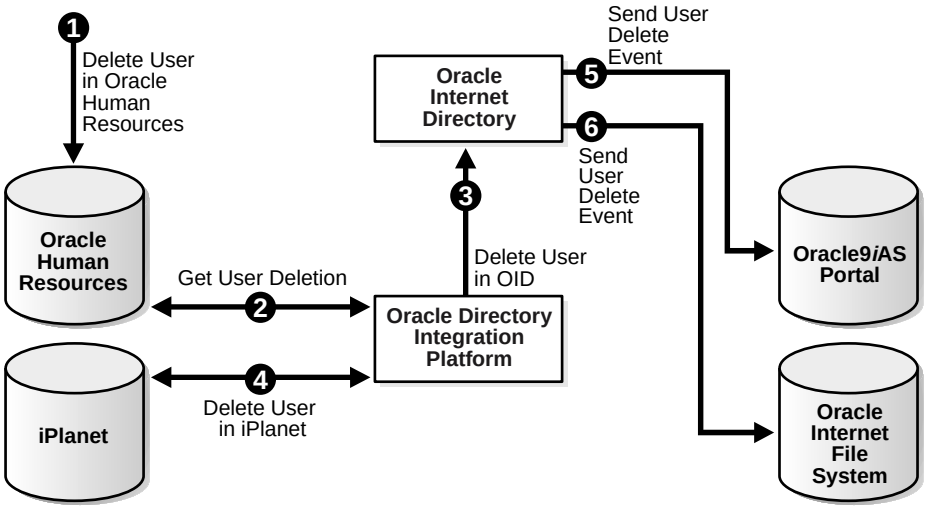
1. The user is first modified in the Oracle Human Resources.
2. The Oracle Directory Integration Platform gets these changes through the synchronization integration service.
3. The Oracle Directory Integration Platform then makes the corresponding user modification in Oracle Internet Directory.
4. The synchronization integration service of the Oracle Directory Integration Platform also modifies the user in the iPlanet Directory Server.
5. The provisioning integration service of the Oracle Directory Integration Platform notifies Oracle9*i*AS Portal about the change in user properties.
6. The provisioning integration service of the Oracle Directory Integration Platform also notifies Oracle Internet File System about the same change in user properties.

Deletion of Users

In this example, the enterprise requires that a user being deleted or terminated in Oracle Human Resources should be automatically be denied access to all enterprise

resources that are based on the directory service. The following figure shows the flow of events during the deletion of users:

Figure 28–7 Deletion of Users from the Corporate Human Resources



As Figure 28–7 shows, the process by which Oracle Directory Integration Platform communicates the deletion of users to all systems in the enterprise contains the following sequence of events:

1. The user is first deleted in the Oracle Human Resources.
2. The Oracle Directory Integration Platform gets these changes through the synchronization integration service.
3. The Oracle Directory Integration Platform then makes the corresponding user deletion in Oracle Internet Directory.
4. The synchronization integration service of the Oracle Directory Integration Platform also deletes the users in the iPlanet Directory Server.
5. The provisioning integration service of the Oracle Directory Integration Platform notifies Oracle9*i*AS Portal about the deletion of the user.
6. The provisioning integration service of the Oracle Directory Integration Platform also notifies Oracle Internet File System about the deletion of the user.

Once all of the steps indicated above are completed, a deleted user in Oracle Human Resources can no longer access any corporate services like Oracle9*i*AS Portal or Oracle Internet File System.

29

The Oracle Directory Synchronization Service

This chapter discusses Oracle Directory Synchronization Service, which uses the first of the two types of integration profiles: the "directory synchronization profile." This profile provides the configuration information necessary to make Oracle Internet Directory and connected directories consistent.

This chapter discusses the synchronization profiles and connectors that link Oracle Internet Directory and connected directories. It contains the following topics:

- About Connectors and Directory Integration Profiles
- Managing Synchronization Profiles
- Managing Connectors from the Command Line

> **See Also:** "Provisioning Integration Service" on page 28-8 for a discussion of the second type of integration profile, called a provisioning integration profile, which identifies the data and methods to be used to notify an application of changes in user or group data

About Connectors and Directory Integration Profiles

This section contains these topics:

Connectors

In the Oracle Directory Integration Platform, a connector represents a prepackaged connectivity solution between Oracle Internet Directory and a connected directory. Minimally, it consists of a connector profile called Directory Integration Profile, which contains all the configuration information required for synchronizing data between Oracle Internet Directory and a connected directory. This is all that one needs for synchronizing between Oracle Internet Directory and a connected directory if the connected directory can support one of the interfaces supported by DIP for exchanging data. One example is the iPlanet Connector that is shipped with the Oracle Internet Directory product. The iPlanet connector consists of just a pre-packaged Integration profile, because the data between Oracle Internet Directory and iPlanet Directory can be synchronized using the LDAP interface supported by Oracle Directory Integration Platform.

A connector may also include an agent. This is required if the connected directory can not directly support the interface supported by DIP for exchanging data. The agent would transform the data from one of the data formats supported by DIP into a format supported by the connected directory. An example is the Oracle HR Connector, which consists of a prepackaged Integration profile and an HR agent. This agent uses the "Tagged File" format supported by DIP to communicate data with Oracle Internet Directory, and it uses SQL (through OCI interface) to communicate with the Oracle Human Resources system.

Directory Synchronization Profiles

A directory integration profile for synchronization is called a directory synchronization profile. It contains all the configuration information required for synchronization—for example, the name and type of an agent, how and when to invoke it, the mapping information required for synchronizing entries and attributes.

Some connected directories only receive data from Oracle Internet Directory, and do not supply data to Oracle Internet Directory. Others supply data to Oracle Internet Directory but do not receive data from Oracle Internet Directory. Some directories both supply data to and receive data from Oracle Internet Directory. A separate profile is used for each direction, that is, for information coming into Oracle Internet Directory and for information going from Oracle Internet Directory to the connected directories.

Some connected directories can receive data in any of the interfaces built into Oracle Internet Directory for synchronization. These interfaces currently include the PL/SQL, LDAP, tagged, and LDIF interfaces. For these connected directories, the Directory Synchronization Service performs the synchronization itself, directly, using the information stored in the profile.

Changes requiring synchronization can occur in Oracle Internet Directory or in a connected directory. The Directory Synchronization Service (DSS) periodically checks each profile, comparing its last successful update time and change number against the contents of the Change Log. When as-yet-unsynchronized changes are found, the DSS initiates synchronization. Import and export operations for Oracle Internet Directory are handled directly by the Oracle Directory Integration Server. If synchronization with a particular connected directory requires use of an agent, that need is specified in the profile and the agent is automatically invoked.

Directories with Unique Formats

Some connected directories cannot receive data using any of those interfaces. The profiles for this type of directory contain an attribute identifying a separate program to be used to accomplish the synchronization. This program, called an agent, translates between the connected directory's specialized format and a tagged or LDIF file containing the synchronization data. The Directory Synchronization Service invokes the agent identified in the profile to perform the synchronization.

When exporting synchronization data from the Oracle Internet Directory for import into this type of connected directory, the Directory Synchronization Service creates the necessary file in the tagged or LDIF format. The agent then reads that file,

translates it into the correct format for the receiving connected directory, and stores the data in that directory.

When exporting synchronization data from this type of connected directory for import into the Oracle Internet Directory, the agent creates the necessary tagged or LDIF format file. The Directory Synchronization Service then uses this file of connected directory data to update the Oracle Internet Directory.

Synchronization Scenarios

Synchronization can occur in either direction, i.e., from a connected directory to Oracle Internet Directory or from Oracle Internet Directory to a connected directory (or both).

Synchronizing from Oracle Internet Directory to a Connected Directory

A numbered entry is stored in the Change Log Container for each change to Oracle Internet Directory. Each time the Directory Synchronization Service processes a synchronization profile, it retrieves the number of the Change Log entry last used to update the corresponding connected directory. Checking each Change Log entry after (more recent than) that number, the Service uses the profile's filtering rules to select changes requiring synchronization with the corresponding connected directory.

The appropriate entries or attributes are then updated in that connected directory. (If it does not use PL/SQL, LDAP, tagged, or LDIF formats directly, then the connector identified in its profile is invoked.) The last Log number successfully used is then stored in the profile.

Oracle Internet Directory periodically purges the Change Log after all profiles have used what they need, identifying where subsequent synchronization should begin.

Synchronizing from a Connected Directory to Oracle Internet Directory

When a connected directory uses PL/SQL, LDAP, tagged, or LDIF formats directly, changes to its entries or attributes are automatically synchronized by the Directory Synchronization Service. Otherwise, the connector identified in its synchronization profile must write the changes to an export file in tagged or LDIF format. The Directory Synchronization Service then uses this file of connected directory data to update the Oracle Internet Directory.

Registration of Connectors into Oracle Directory Integration Platform

Before deploying a connector, you register it in Oracle Internet Directory. This registration involves creating a directory synchronization profile in the directory. This synchronization profile is stored as an LDAP entry in the directory. To create it, you can use either Oracle Directory Manager or command-line tools, as described in subsequent sections of this chapter.

Most of the information needed to synchronize the data with the connected directory—such as accountname, password, hostname, portnumber—is stored in the synchronization profile. However, if the connector execution requires any additional information, it can be stored in the `orclOdipAgentConfigInfo` attribute discussed in the section "Additional Connector Configuration Information" later in this chapter.

Attributes in a synchronization profile entry belong to the object class `orclodiProfile`. The only exception is the `orcllastChangeLogNumber` attribute, which belongs to the object class `orclChangeSubscriber`.

The Object ID prefix `2.16.840.1.113894.7` is assigned to platform-related classes and attributes. The following table lists all the attributes in the Oracle Directory Integration Platform profile.

Table 29–1 Attributes in the Oracle Directory Integration Platform Profile

Attribute	Description
General Information	
ProfileName (`orclOdipAgentName`)	Name of the Integration Profile.
ProfileStatus (`orclOdipAgentControl`)	Indicator whether the profile is enabled or disabled.
Profile Password (`orclOdipProfilePassword`)	The password used by the profile to bind to Oracle Internet Directory. In case of import, the changes are made as with profilename as the identity.
SynchronizationMode (`orclOdipSynchronizationMode`)	IMPORT/EXPORT. Import implies changes from the connected directory are imported to Oracle Internet Directory. Export implies changes from the Oracle Internet Directory are extracted and given to the connected directory.
SchedulingInterval (`orclOdipSchedulingInterval`)	The interval with which the connector has to synchronize.

Table 29–1 Attributes in the Oracle Directory Integration Platform Profile

Attribute	Description
Number of Retries (`orclodipSyncRetryCount`)	Maximum number of times the agent or synchronization will be attempted in case of failure. By default, the Directory Integration Server tries the synchronization a maximum of 5 times. The first retry takes place 1 minute after the first failure, 2nd retry happens 2 minutes after the 2nd failure and subsequently the n-th retry takes place after n minutes after the n-th failure.
ProfileVersion (`orclVersion`)	Identifier indicating the Integration Profile version. It has a value 1.0. If this field has a value other than 1.0, the profile will not be processed.
Execution Information	
AgentExecutionCommand (`orclodipAgentExeCommand`)	Connector executable name and argument list used by the directory integration server. It can be passed as a command-line argument when the connector is invoked. Typical usage of passing it in the command-line is illustrated in Chapter 33, "Synchronization with Oracle Human Resources".
ConnectedDirectory Account (`orclOdipConDirAccessAccount`)	Valid user account in the connected directory to be used by the connector for synchronization. For instance, for the Iplanet Synchronization Connector, it is the valid binddn in the iPlanet directory. For Hragent, it is a valid user id in the HR database. For other connectors, it can be passed as a commandline argument when the connector is invoked. Typical usage of passing it in the commandline is illustrated in Chapter 33, "Synchronization with Oracle Human Resources".
ConnectedDirectory AccountPassword (`orclOdipConDirAccessPassword`)	Password to be used by the userid specified by 'ConnectedDrectoryAccount' to connect to the connected directory. For instance, for the Iplanet Synchronization Connector, it is the valid bindpassword in the iPlanet directory. For Hragent, it is the HR Database password.

Table 29–1 Attributes in the Oracle Directory Integration Platform Profile

Attribute	Description
Connected Directory URL (orclOdipConDirURL)	Connect details required to connect to the connected directory. In the case of iPlanet Synchronization, this parameter refers to the hostname and portnumber as, "host:port". Similarly for DB this can be used in the form of 'Host:port:oraclesid'.
Interface Type (orclodipDataInterfaceType)	The data format or protocol used in synchronization. The four supported values are: 1. LDIF - Import/Export from a LDIF File 2. Tagged - Import/Export from a Tagged File - a proprietary format supported by the Integration server, similar to LDIF format The details are discussed in Appendix A, "Syntax for LDIF and Command-Line Tools". 3. LDAP - Import/Export of the data from/to a LDAP compliant directory. 4. DB - Import/Export of the data from/to a RDBMS directory.
Additional Config Info (orclOdipAgentConfigInfo)	Any additional configuration Information that needs to be passed onto the connector. When the connector is scheduled for execution, the value of the attribute is stored in the file, '$ORACLE_HOME/ldap/odi/conf/profilename.cfg' which can be processed by the connector.
Mapping Information	
Attribute Mapping Rules (orclOdipAttributeMappingRules)	Mapping rules for converting data from a connected directory to Oracle Internet Directory. This information is stored as a binary attribute. Mapping rules are discussed in greater detail in Mapping Rules and Formats on on page 29-9. **See Also**: "Default Oracle Human Resources Connector Mapping Rules" on page 33-13 for an example of mapping rules.
ConnectedDirectoryMatchingFilter (orclOdipConDirMatchingFilter)	Attribute used to filter changes made to Oracle Internet Directory to select those to be applied to the connected directory.

Table 29–1 Attributes in the Oracle Directory Integration Platform Profile

Attribute	Description
OIDMatchingFilter (`orclOdipOIDMatchingFilter`)	Attribute used to filter changes made to the connected directory to select those to be applied to Oracle Internet Directory.
Status Information	
LastExecutionTime (`orclOdipLastExecutionTime`)	Time when synchronization was last carried out. Its format is dd-mon-yyyy hh:mm:ss, where hh is the time of day in a 24 hour format.
LastSuccessfulExecutionTime (`orclOdipLastSuccessfulExecutionTime`)	Time of the last successful synchronization, in the format dd-mon-yyyy hh:mm:ss, where hh is the hour in 24-hour format.
Synchronization Status (`orclOdipSynchronizationStatus`)	Synchronization status of the last execution: Success / Failure.
SynchronizationError (`orclodipSynchronizationErrors`)	Reason for failure (if last execution failed)
Con Dir Last Applied Change Num (`orclodipConDirLastAppliedChgNum`)	For import operations, the last change from the connected directory that was applied to Oracle Internet Directory.
OIDLastAppliedChangeNumber (`orclOdipLastAppliedChgNum`)	For export operations, the last change from Oracle Internet Directory that was to the connected directory

The various synchronization profile entries in the directory are created under the container `cn=subscriber profile, cn=changelog subscriber, cn=oracle internet directory`.

For example, a connector called OracleHRAgent is stored in the directory as `orclodipagentname=OracleHRAgent,cn=subscriber profile,cn=changelog subscriber,cn=oracle internet directory`.

Additional Connector Configuration Information

While the synchronization profile stores most of the information needed by a connector to synchronize Oracle Internet Directory data with connected directories, some connectors may need more. Some operations might require additional configuration information at runtime.

You can store such additional connector configuration information wherever and however you want. However, the Oracle Directory Integration Platform enables you to store it in the synchronization profile as an attribute called `orclODIPAgentConfigInfo`. Its use is optional: if a connector does not require such information, then the corresponding attribute in the synchronization profile is simply left empty. If such information would be useful, you can load it into this attribute using the script named ldapUploadAgentFile.sh. The type and format of the data stored in the additional configuration information attribute are determined by each executable's needs.

This configuration information can pertain to the connector or to the connected directory or both. Oracle Internet Directory and Oracle directory integration server do not read or modify this information. When the connector is invoked, the Oracle Directory Integration Server simply passes to it the information in this attribute, as a temporary file.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

See Also:

- "Location and Naming of Files" on page 29-18 for the names of these files

Mapping Rules and Formats

In a directory synchronization environment, a typical set of entries from one domain can be moved to another domain. Similarly, a set of attributes can be mapped onto another set of attributes.

Mapping rules govern the conversion of attributes between a connected directory and the Oracle Internet Directory. Each connector has a set of mapping rules stored in the `orclodipAttributeMappingRules` attribute of its synchronization profile.

The Oracle directory integration server uses these rules to map attributes as needed when exporting data from the directory and interpreting the data imported from a connected directory or file. When the Directory Integration Server imports changes

into the Oracle Internet Directory, it converts the connected directory's change record into an LDAP change record following the mapping rules. Similarly, during export, the connector translates Oracle Internet Directory changes to the format understood by the connected directory.

The Mapping Rules attribute provides a means of specifying domain level mapping and attribute level mapping. It can be assumed to be in the format of a file as described below:

Mapping rules are organized in a fixed tabular format, and you must follow that format carefully. Each set of mapping rules appears between a line containing only the word DomainRules and a line containing only the characters "###" (without the quotes). The fields within each rule are delimited by a colon (:).

- **DomainRules**

  <srcDomainName1>: [dstDomainName1]: [DomainMappingRule1]
  <srcDomainName2>: [dstDomainName2]: [DomainMappingRule2]

- **AttributeRules**

  <srcAttrName1>: [ReqAttrSeq]:[SrcAttrType]: [SrcObjectClass]:[dstAttrName1]:
  [DstAttrType]: [DstObjectClass]:[AttrMappingRule1]
  <srcAttrName2>: [ReqAttrSeq]:[SrcAttrType]: [SrcObjectClass]:[dstAttrName2]:
  [DstAttrType]: [DstObjectClass]:[AttrMapping Rule2]
  ###

  where the expansion of each <srcAttrName1> and <srcAttrName2> would each be a single unfolded long line.

The domain rule specifications appear after a line containing only the keyword `DomainRules`. Each domain rule is represented with the components (separated by colons) that are described in Table 29–2, "DomainRule Components".

Table 29–2 DomainRule Components

Component Name	Meaning and Use
SrcDomainName	This entry gives the name of the domain/container of interest. Specify NONLDAP for sources other than LDAP and LDIF.
DstDomainName	Name of the domain of interest in the destination. It is optional, and if not specified, takes the value of <SrcDomainName> under valid conditions. For destinations other than LDAP and LDIF, specify NONLDAP. Since import/export always refers to Oracle Internet Directory, a combination of NONLDAP:NONLDAP is not allowed.
DomainMappingRule	This field is meaningful only in import to Oracle Internet Directory, or in export to LDIF file or to another external LDAP directory. This rule is used for constructing the destination dn from the source domain name and/or the attribute given in AttributeRules. This field is typically of the form cn=%,l=%,o=oracle,dc=com. Such specifications are used to put entries under different domains or containers in the directory. In case of Non-LDAP sources, this rule indicates the way the target dn needs to be formed to place the entries in the directory. This component is optional in LDAP to LDIF, LDAP to LDAP, or LDIF to LDAP. If not specified, the source domain and destination domain names are considered to be the same.

The attribute rule specifications appear after a line containing only the keyword `AttributeRules`. Each attribute rule is represented with the components (separated by colons) that are described in Table 29–3 on page 29-12. The attribute rule specifications end with a line containing only the characters "###" (without the quotes).

Table 29–3 Components in Attribute Rules

Component	Discussion
SrcAttrName	For LDAP directory repositories, this parameter refers to the name of the attribute to be translated. For RDBMS repositories, it refers to the ColumnName in the table specified by the SrcClassName. For other repositories this parameter can be appropriately interpreted.
ReqAttrSeq	This field indicates whether the source attribute must always be passed on to the destination. When entries are synchronized between the directory and the connected directory, some attributes need to be used as synchronization keys. This field is to indicate whether the specified attribute is being used as a key. If so, irrespective of whether the attribute has changed or not, the value of the attribute is always extracted from the source. A non-zero integer value should be placed in this field if the attribute needs to be always passed on to the other end.
SrcAttrType	This parameter refers to the attribute type (integer, String, binary, etc.), which will be helpful in validating the mapping rules, i.e., validating the equivalency of the Source and Destination attribute types. (In the current Release, this field is ignored.)
SrcObjectClass	If the source of the attribute being shared is an LDAP directory, this parameter names the object class to which the attribute belongs. If the source of the attribute being shared is an RDBMS repository, this parameter refers to the TableName. (This specification is mandatory for LDAP or RDBMS.) For other repositories, this parameter may be ignored.
DstAttrName	This is optional. If it is not specified, the SrcAttrName will be assumed. For LDAP directory repositories, this parameter refers to the name of the attribute at the destination. For RDBMS repositories, it refers to the ColumnName in the table specified by the SrcClassName. For other repositories, this parameter can be appropriately interpreted.
DstAttrType	This parameter refers to the attribute type (integer, String, binary, etc.), which will be helpful in validating the mapping rules, i.e., validating the equivalency of the Source and Destination attribute types. (In the current Release, this field is ignored.)

Table 29–3 Components in Attribute Rules

Component	Discussion
DstObjectClass	For LDAP directory repositories, this parameter refers to the object class to which the attribute belongs, and is optional. For RDBMS repositories, it refers to the TableName, and is mandatory. For other repositories this parameter may be ignored.
AttrMapping Rule	Optional arithmetic expression with operators: +, functions: toUpper (string) , toLower(String), trunc (string,char). If nothing is specified, the source attribute value is copied as the value of the destination attribute.

OrclodipAttributeMappingRules is a single valued attribute in the directory. It needs to follow a fixed format. Therefore, editing the mapping rules in ODM is not feasible.

To overcome this, mapping rules are stored in a file, and the file is uploaded to the directory as a value of the attribute. The utility ldapUploadAgentFile.sh can be used for uploading the mapping file.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

In a newly created synchronization profile, mapping rules will be empty. To enter mapping rules, edit a file which strictly follows the format given in the previous section.

Here is a sample mapping file that can be used to import HR data from the Oracle HR Database tables using TaggedFile Interface. (This file is supplied during installation, at $*ORACLE_HOME*/ldap/odi/conf/oraclehragent.map.master.)

```
DomainRules
NONLDAP:dc=metaagt,dc=com:uid=%dc=metaagt,dc=com
AttributeRules
firstname: : : :cn: :person
email : : : :cn: :person: trunc(email,'@')
```

```
email : : : :uid: :person:trunc(email,'@')
firstname,lastname: : : :cn: :person: firstname+","+lastname
lastname,firstname: : : :cn: :person: lastname+","+firstname
firstname,lastname: : : :sn: :person: lastname | firstname
EmployeeNumber: : : :employeenumber: :inetOrgperson
EMail: : : :mail: :inetOrgperson
TelephoneNumber1: : : :telephonenumber: :person
TelephoneNumber2: : : :telephonenumber: :person
TelephoneNumber3: : : :telephonenumber: :person
Address1: : : :postaladdress: :person
state: : : :st: :locality
street1: : : :street: :locality
zip: : : :postalcode: :locality
town_or_city: : : :l: :locality
Title: : : :title: :organizationalperson
#Sex: : : :sex: :person
###
```

As described earlier, the mapping file consists of keywords and a set of domain and attribute mapping rule entries. The following explanations may help you understand the sample file. It contains the domain rule `NONLDAP:dc=metaagt,dc=com:cn=%,dc=metaagt,dc=com`. This rule implies that the source domain is NonLDAP, indicating there is no source domain.

The destination domain (`:dc=metaagt,dc=com`) implies that all the directory entries this profile deals with are in the domain dc=metaagt,dc=com.

The DomainMappingRule (`: uid=%,dc=metaagt,dc=com`) implies that the data from the source should refer to the entry in the directory with the dn, which is constructed using this domain mapping rule. In this case, 'uid' must be one of the destination attributes which should always have a non-null value. If any data corresponding to an entry to be synchronized has a 'null' value, then the mapping engine assumes that the entry is invalid and proceeds to the next entry. To identify the entry correctly in the directory, it is also necessary that 'uid' should be a single-valued attribute.

In some cases, the 'rdn' of the 'dn' needs to be constructed using the name of a multivalued attribute. For example, to construct an entry with the 'dn' of

'cn=%,l=%,dc=metaagt,dc=com', where 'cn' is a multi-valued attribute, the DomainMappingRule can be of this form: `rdn,l=%,dc=metaagt,dc=com`

where rdn is one of the destination attributes having a non-null value. A typical mapping file supporting this could have the following form:

```
DomainRules
NONLDAP:dc=metaagt,dc=com:rdn,l=%,dc=metaagt,dc=com
AttributeRules
firstname: : : :cn: :person
email : : : :cn: :person: trunc(email,'@')
email : : : :rdn: :person: 'cn='+trunc(email,'@')
firstname,lastname: : : :cn: :person: firstname+","+lastname
lastname,firstname: : : :cn: :person: lastname+","+firstname
firstname,lastname: : : :sn: :person: lastname | firstname
EmployeeNumber: : : :employeenumber: :inetOrgperson
EMail: : : :mail: :inetOrgperson
TelephoneNumber1: : : :telephonenumber: :person
TelephoneNumber2: : : :telephonenumber: :person
TelephoneNumber3: : : :telephonenumber: :person
Address1: : : :postaladdress: :person
Address1: : : :postaladdress: :person
Address1: : : :postaladdress: :person
state: : : :st: :locality
street1: : : :street: :locality
zip: : : :postalcode: :locality
town_or_city: : : :l: :locality
Title: : : :title: :organizationalperson
#Sex: : : :sex: :person
###
```

In the attribute mapping rule, `firstname: : :cn: : person`, these explanations apply:

SrcAttrName - firstname (Name of the original attribute)

ReqAttrSeq : empty (If the attr is not found, you can still continue with mapping)

SrcAttrType: empty (Not required)

SrcObjectClass: empty (Not required)

DstAttrName : cn (Name of the attr as it appears in Oracle Internet Directory)

DstAttrType: empty (Not required)

DstObjectClass : person. Objectclass to which the attribute belongs to - it is mandatory while using a Import with Tagged File interface.

Similarly, the rule `email: : :cn: : person: trunc(email,'@')`

implies applying the mapping rule of truncating all the characters off of 'email' and get the remaining as 'cn'.

You can customize mapping rules by adding new ones, modifying the existing ones or deleting the existing ones by modifying the file. If the mapping rules are not available in a file, the attribute value can be downloaded to the file using ldapsearch. For usage of the ldapsearch command, see Appendix A. The entry to be searched for is '`orclodipagentname=<ProfileName>,cn=subscriber profile,cn=changelog subscriber,cn=oracle internet directory`' for the attribute '`orclodipattributemappingrules`'.

The Oracle Directory Integration Platform supports both one-to-many and many-to-one mappings.

- One-to-many mapping

  One attribute in a connected directory can map to many attributes in Oracle Internet Directory. For example, suppose an attribute in the connected directory is `Address:123 Main Street/MyTown, MyState 12345`. You can map this attribute in Oracle Internet Directory to both the LDAP attribute `homeAddress` and the LDAP attribute `postalAddress`.

- Many-to-one mapping

  Multiple attributes in a connected directory may map to one attribute in Oracle Internet Directory. For example, suppose that the Human Resources directory represents Anne Smith by using two attributes: `firstname=Anne` and `lastname=Smith`. You can map these two attributes to one attribute in Oracle Internet Directory: `cn=Anne Smith`.

See Also: "Default Oracle Human Resources Connector Mapping Rules" on page 33-13 for an example of mapping rules

Updating Mapping Rules

You can customize mapping rules by adding new ones, modifying existing ones, or deleting some from the mapping rule set specified in the `orclodipAttributeMappingRules` attribute. In general, to perform any of these operations, you identify the file containing the mapping rules or store the value of the attribute for a file using an ldapsearch command as described in Appendix A, "Syntax for LDIF and Command-Line Tools".

`orclodipAttributeMappingRules` is a single-valued attribute in the directory, which needs to follow a fixed format. Hence editing the mapping rules in ODM is not feasible. To overcome this, mapping rules are stored in a file that is uploaded to the directory as a value of the attribute. The utility ldapUploadAgentFile.sh can be used to do this. Once the mapping file is created and uploaded, a copy of the file can be maintained in the $*ORACLE_HOME*/`ldap/odi/conf` directory, and uploaded again after any future update.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Adding an Entry to the Mapping Rules File To add a new entry to the mapping rules file, edit this file and add a record to it. To do this:

1. Identify the connected directory attribute name that needs to be mapped to Oracle Internet Directory.
2. Identify the corresponding attribute name in Oracle Internet Directory to which it can be mapped.
3. Generate the mapping rule elements indicating the conversion that needs to be done on the attribute values.
4. Load the attribute mapping rule file to the synchronization profile by using the ldapUploadAgentFile.sh tool.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Modifying an Entry in the Mapping Rules File After you identify an entry to be modified in the mapping rules file, generate the mapping rule element for the desired conversion of attribute values. Then use the ldapUploadAgentFile.sh tool to load the attribute mapping rule file into the synchronization profile.

Deleting an Entry from the Mapping Rules File After you identify an entry to be deleted in the mapping rules file, you can either delete the entry from the file or comment it out by putting a hash mark (#) in front of it. Then use the ldapUploadAgentFile.sh tool to load the attribute mapping rule file into the synchronization profile.

Location and Naming of Files

Table 29–4 tells you where to find the various files and what names to use:

Table 29–4 Location and Names of Files

File	File Name
Import DataFile	$ORACLE_HOME/ldap/odi/data/import/ProfileName.dat
Export Data File	$ORACLE_HOME/ldap/odi/data/export/ProfileName.dat
TraceFile	$ORACLE_HOME/ldap/odi/log/ProfileName.trc
Additional Configuration Info	$ORACLE_HOME/ldap/odi/conf /ProfileName.cfg
Mapping Rules	$ORACLE_HOME/ldap/odi/conf /ProfileName.map

For example, the datafile name of the Oracle Human Resources agent is `oraclehrprofile.dat`.

Managing Synchronization Profiles

This section contains these topics:

Managing Profiles by Using Oracle Directory Manager

This section tells you how to register and deregister a profile by using Oracle Directory Manager.

Registering a Profile by Using Oracle Directory Manager

Oracle Directory Manager enables you to register a profile in one of two ways:

- By creating a new configuration set entry, then adding a profile to it
- By selecting an existing configuration set entry, then adding a profile to it

To register a profile:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Server Management, then select Directory Integration Server. The Active Processes box appears in the right pane.
2. On the toolbar, click Create. The Configuration Sets dialog box appears.
3. In the Configuration Sets dialog box, click Create. The Integration Profiles dialog box appears. You have two options:
   - To create an integration profile by copying an existing one, select the Oracle Directory Integration Platform profile you want to copy, then click Create Like. The Integration Profile dialog box displays the General tab page.
   - To create an integration profile without copying an existing one, click Create New. The Integration Profile dialog box displays the General tab page.

4. In the General tab page, fill in the fields as explained in Table 29–5.

Table 29–5 Description of Fields on the General Tab Page in Oracle Directory Manager

Field	Description
Profile Name	Specify the name of the Profile. The name you enter is used as the RDN component of the DN for this integration profile. For example, specifying a profile name `MSAccess` creates an integration profile named `orclodipagentname`=MSAccess,cn=subscriber profile, cn=changelog subscriber, cn=oracle internet directory. This field is mandatory. There is no default.
Synchronization Mode	Specify whether this is an import or an export operation. An import operation pulls changes from a connected directory into Oracle Internet Directory. An export operation pushes changes from Oracle Internet Directory into a connected directory. This field is mandatory. The default is `IMPORT`.
ProfileStatus	Specify whether the profile is enabled or disabled. This field is mandatory. The default is ENABLE.
Number of Retries	Specify the maximum number of times the directory integration server is to attempt synchronization before it disables synchronization. This field is mandatory. The default is 5. The first retry takes place 1 minute after the first failure. The 2nd retry happens 2 minutes after the 2nd failure, and subsequently the n-th retry takes place n minutes after the n-th failure.
Scheduling Interval	Specify the number of seconds between synchronization attempts between a connected directory and Oracle Internet Directory. This field is mandatory. The default is `60`.
Agent Execution Host	The host on which the agent is to be executed.

5. Select the Execution tab and fill in the fields as explained in Table 29–6.

Table 29–6 Description of Fields on the Execution Tab in Oracle Directory Manager

Field	Description
Execution Command	Specify the agent executable name and the arguments used by the directory integration server to execute the agent. This field is optional. There is no default. A typical execution command is of the form, `odicmd user=%orclodipcondirAccessAccount pass=%orclodipcondiraccesspassword` Where `odicmd` is the command to be executed (available in the PATH or specified as a complete pathname), and `user=%orclodipcondirAccessAccount pass=%orclodipcondiraccesspassword` are the commandline arguments. The value to be passed for the user is derived from the attribute `orclodipcondiraccessaccount` and the value to be passed for 'pass' is derived from the attribute `orclodipcondiraccesspassword`. A typical example is given in HR agent.
Connected Directory Account	Specify the account to be used by the connector / agent for accessing the connected directory. For example, if the connected directory is a database, the account might be `Scott`. If the connected directory is another LDAP-compliant directory, then the account might be `cn=Directory Manager`. This field is optional. There is no default.
Connected Directory Account Password	Specify the password the connector / agent is to use when accessing the connected directory. This field is optional. There is no default.
Additional Config Info	This field displays additional information that the directory integration server passes to an agent. You cannot modify this field through ODM. The only way to modify it is to use ldapuploadagentfile.sh. There is no default.
Connected Directory URL	The URL of the connected directory, if available.
Data Interface Type	The format used by the import or export file. Valid values are `LDIF, DB, LDAP,` or `TAGGED`. This field is optional. The default is `TAGGED`.

6. Select the Mapping tab and fill in the fields as explained in Table 29–7.

Table 29–7 Description of Fields on the Mapping Tab in Oracle Directory Manager

Field	Description
Mapping Rules	This field displays the mapping rules for converting data between a connected directory and Oracle Internet Directory. There is no default. **Note:** You cannot edit the mapping rules file by using Oracle Directory Manager. You edit the mapping rules in a file manually and then upload it to the `profile` by using the provided script, `ldapUploadAgentFile.sh`. See Appendix A, "Syntax for LDIF and Command-Line Tools"
OIDMatchingRule	Specify the attribute that uniquely identifies records in Oracle Internet Directory. This attribute is used as a key to synchronize Oracle Internet Directory and the connected directory. This field is optional.
ConnectedDirectorymatchingRule	Specify the attribute that uniquely identifies an entry in the connected directory.

7. Select the Status tab and fill in the fields as explained in Table 29–8. Since this shows the execution status of the connectors, most of the fields are not editable.

Table 29–8 Description of Fields on the Status Tab in Oracle Directory Manager

Field	Description
OID Last Applied Change Number	For export operations, specify the identifier of the last change from Oracle Internet Directory that has been applied to the connected directory. The default is `0`. The field can be consciously modified by the end user whenever appropriate. The profile should be in the disabled mode. If the number is increased, then any Change Log entries numbered between the original value and the new value will not be applied.
Last Execution Time	The most recent absolute time that the agent was executed. The default is the time at which the connector is created. Modifying this field will be misleading.
Last Successful Execution Time	The most recent absolute time that the agent succeeded. The default is the time at which the connector is created. Modifying this field will be misleading.
Synchronization Status	Synchronization success / failure.

Table 29–8 Description of Fields on the Status Tab in Oracle Directory Manager

Field	Description
Synchronization Errors	The last error message. You cannot modify this field. There is no default.
ConnectedDirectory Last AppliedChangeNumber	The number of the Change Log entry that was most recently applied successfully to the connected directory. The field can be consciously modified by the end user whenever appropriate. The profile should be in the disabled mode. If the number is increased, then any Change Log entries numbered between the original value and the new value will not be applied.

8. When all edits under every tab of the Integration Profile dialog box are completed, click OK. This returns you to the Configuration Sets dialog box, which now lists the integration profile you just created.
9. Click OK to exit the Configuration Sets dialog box. The agent you created is now registered with Oracle Internet Directory.

Deregistering a Profile by Using Oracle Directory Manager

To delete a connector:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance*> Server Management > Directory Integration Server.
2. Select the Configuration Set from which to delete the agent. The Integration Profiles tab page appears in the right pane.
3. In the Integration Profiles tab page, select the agent you want to deregister, then click Delete.

Managing Connectors from the Command Line

This section tells you how to register and deregister agents by using the script `ldapcreateConn.sh`.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Creating a Synchronization Profile with the Command-Line Tool

You can create a synchronization profile by using the command-line tool ldapcreateConn.sh. This tool is in the directory $*ORACLE_HOME*/ldap/admin/. The lines below show the syntax for this tool; Table 29–9 explains its arguments.

```
LdapcreateConn.sh -name <Agent Name> \
  [ -type <IMPORT | EXPORT > ] [ -agentpwd < Agent Password> ] \
  [ -config <which configset to associate to > ] \
  [ -LDAPhost <LDAP server host> ] \
  [ -LDAPport <LDAP server port> ] \
  [ -binddn SuperUserDN (default cn=orcladmin ) ] \
  [ -bindpass   Bindpassword (default=welcome) ] \
  [ -retry   <Max Retry Count on synchronization Errors > ]\
  [ -poll < Polling Interval For Synchronization> ] \
  [ -host < Host on which to run Agent> ]
  [ -conndirurl  < Connected Directory URL > ] \
  [ -conndiracct < Connected Directory Acct Info > ]\
  [ -conndirpwd < Connected Directory Acc Pwd> ] \
  [ -execmd < Command Line for the Agent > ]\
  [ -iftype < Interface Type > ]      \
  [ -condirfilter < Connected Directory Matching Filter> ]\
  [ -oidfilter < OID Matching Filter > ] \
  [ -U <SSL Authentication Mode> ]\
  [ -W <Wallet location> ]   [ -P <Wallet password> ]
```

Table 29–9 Arguments for ldapcreateCoArguments for ldapcreateConn.sh to Register a Partner Agent

Argument	Description
Name	The Name of the Integration Profile.This has to be unique.
Type	IMPORT/EXPORT. The default is IMPORT/
Agentpwd	The password to protect the profile. The default is 'welcome'.
Config	The configuration set number. The default is 1.
LDAPhost	The LDAP Server host. The default is the current host.
LDAPport	The LDAP server port The default is port 389.
Binddn	The BIND DN of the Directory user which has the privileges to create Integration profile. The default is 'cn=orcladmin'
Bindpass	The BIND password. The default is 'welcome'
Retry	Maximum number of retries to be done by the server on encountering a synchronization error. The default is '5'.
Poll	The scheduling interval of the profile. The default is '60' seconds.
Host	This is currently used. For the time being, it should be set to the machine name on which the DIP server is executing.
Conndirurl	The Connected Directory access Information.
Conndiracct	The Connected Directory account.
Conndirpwd	The Connected Directory account password
Execmd	The OS command line to execute the partner agent.
Iftype	The Interface Type. The default is TAGGED.
Condirfilter	The Connected Directory Matching Filter
Oidfilter	The Oracle Internet Directory Matching Filter.

When the integration server is invoked with configuration set 2 in this command line argument, this agent is run. You can see a full description by invoking ldapCreateConn.sh with the `-help` argument.

Deregistering a Profile Using ldapdeleteConn.sh

You can deregister a agent by using the command-line tool `ldapdeleteConn.sh`. This tool is in the directory `$ORACLE_HOME/ldap/admin/`.

The following example deregisters an agent entry and dissociates it from the configuration set 2 (`config 2`) entry:

```
ldapdeleteconn.sh name HRMS config 2
```

30

Oracle Directory Integration Server Administration

This chapter discusses Oracle directory integration server and tells you how to configure and manage it. It contains these topics:

What the Oracle Directory Integration Server Is

The Oracle directory integration server is the central component of the Oracle Directory Integration Platform. It is a server process that does the following:

- Schedules connectors

  The directory integration server controls the time of data synchronization between OID and connected directories. If there is an agent, its execution time is also scheduled. All such scheduling information for each such directory is stored in its synchronization profile.

- Data import and export

  The directory integration server imports and exports changes into and out of Oracle Internet Directory. LDIF, LDAP, and tagged interfaces are supported.

- Mapping

  The directory integration server includes a generic facility for filtering and mapping data to the connected directories. The directory integration server maps attributes while exporting data to a connected directory and when interpreting import data from a file or directory for input to Oracle Internet Directory.

You can run multiple directory integration server instances on any host.

Registering the Oracle Directory Integration Server

After installing the directory integration server, you must register it with Oracle Internet Directory by using Oracle directory integration server registration tool (`odisrvreg`). You must separately register each directory integration server installed on a different host, by running `odisrvreg` on that host. To run this tool, you need the privileges of an Oracle Internet Directory administrator.

As part of registration, the tool creates an entry in the directory. It sets the password for the directory integration server and stores it as an encrypted value in the registration entry. If the registration entry already exists, then you can use the tool to reset the existing password. The `odisrvreg` tool also creates a local file called `odisrvwallet`, at `$`*`ORACLE_HOME`*`/ldap/odi/conf`. This file acts as a private wallet for the directory integration server, which uses it on startup to bind to the directory.

Note: During installation, the Oracle directory integration server is registered by using the default password of `welcome`, and the directory integration server is running. This is also true if you are upgrading from an earlier version. Oracle Corporation recommends that you change this password and restart the Oracle directory integration server.

Table 30–1 describes the parameters `odisrvreg` uses. You can also run `odisrvreg` in SSL mode to make communication between the tool and the directory fully secure, using three additional parameters that are also in Table 30–1.

To register the directory integration server in non-SSL mode, enter this command:

```
odisrvreg -h hostname -p port -D binddn  -w bindpasswd
```

Table 30–1 Descriptions of ODISRVREG Arguments

Argument	Description
`-h` *`hostname`*	Oracle directory server host name
`-p` *`port_number`*	Port number on which the directory server is running
`-D b`*`inddn`*	Bind DN. The bind DN must have authorization to create the registration entry for the directory integration server.
`-w` *`bindpasswd`*	Bind password
`-U` *`ssl mode`*	SSL mode. For no authorization, specify `0`. For one-way authorization, specify `1`.

`-W` *`wallet`*	SSL wallet. Enter the full path. For example, on Solaris, you could set this parameter as follows:
	`file:/home/my_dir/my_wallet`
	On Windows NT, you could set this parameter as follows:
	`file:C:\my_dir\my_wallet`

`-P` *`wallet_password`*	Password for opening the SSL wallet

To register Oracle directory integration server in SSL mode, i.e., to run the registration tool in SSL mode, enter the following:

```
odisrvreg -h hostname -p port -D binddn  -w bindpasswd
                -U ssl_mode -W wallet -P wallet_password
```

The three additional parameters, shown here on a separate line for clarity, are actually used on the same command line as the others.

Operational Information about the Oracle Directory Integration Server

This section introduces structural and operational information about the directory integration server and contains these topics:

- The Oracle Directory Integration Server and Configuration Set Entries
- Standard Sequences of Directory Integration Server Events
- Managing Configuration Set Entries

The Oracle Directory Integration Server and Configuration Set Entries

Each directory integration server can execute a set of connections supporting data synchronization between Oracle Internet Directory and connected directories. The set of connectors enabling the server to support these connections is listed in its configuration set and passed as one of the command-line arguments to the server.

Whenever a connector is scheduled to do synchronization, the directory integration server starts up a separate thread. This thread opens an LDAP connection to the directory server, then closes the connection before exiting.

The server has three types of threads of execution in the process:

- Main thread

  This is the daemon thread of the Server. It starts up the Scheduler and periodically sends refresh signals to it, to look for changed profiles and to refresh its cache. This thread also looks for the shutdown signal from the Oracle Internet Directory Process Manager (`oidmon`). This signal causes the thread to shut itself down after it sends a signal to the Scheduler to shut down.

- Scheduler thread

  Schedules the connectors for synchronization based on their scheduling interval. On receipt of a refresh signal from the Main Thread, this Scheduler thread refreshes the synchronization profiles to the latest values.

- Connector thread

  Connector threads are spawned by the Scheduler at their individual scheduling intervals. Upon invocation, a connector thread invokes the connector executable

named in the profile and performs the mapping and filtering of the attributes. A connector thread terminates when its synchronization cycle is over.

If no integration profiles are listed for the configuration set, the Oracle directory integration server waits indefinitely until integration profiles are added to that configuration set. (This wait also occurs if integration profiles are configured for the configuration set, but they are all disabled.)

If the configuration set specified in the command line does not exist in the directory, then the Oracle directory integration server logs this information in the log file and exits.

If the configuration set is not specified, then configuration set 0 is assumed, and all the provisioning profiles are considered for scheduling.

See Also:

- "Managing Server Configuration Set Entries" on page 5-2 for more information on configuration set entries
- "The Oracle Directory Synchronization Service" for instructions on enabling and disabling directory integration agents
- "Setting the Debug Level" on page 30-13 for more information about debug levels

Standard Sequences of Directory Integration Server Events

The Oracle directory integration server is the central component of the Oracle Directory Integration Platform. Any specific instance of the Oracle Directory Integration Server supports either provisioning or synchronization. The directory integration server runs as a multi-threaded process while handling the synchronization and provisioning event propagations.

The three threads described in the previous section work together to create these typical process flow sequences:

- Main Thread Process Sequence
- Scheduler Process Sequence
- Connector Process Sequence

Main Thread Process Sequence

1. On startup, the main thread comes up. This is the daemon thread of the server.

2. The daemon thread starts up the scheduler.
3. Checks of the registration of the instance in the directory. If, the instance is not registered, i.e.if the instance is not started up by OIDMON utility, it performs self-registration in Oracle Internet Directory with the config set number and the instance number details.
4. Periodically checks for the refresh time and signals the scheduler to refresh.
5. Periodically checks for the shutdown signal. On receipt of the shutdown signal, signals the Scheduler thread to shutdown.
6. Once the scheduler thread dies, the main thread unregisters and dies.

Scheduler Process Sequence

1. On having started by the Main thread, reads the config set to find the integration profiles to be scheduled.
2. Creates a list of profiles to be scheduled and schedules them based on their scheduling interval.
3. While creating the list of profiles, validates the attributes. If any of the profile attributes have invalid values, the profile is not considered for synchronization or provisioning.
4. On receipt of the refresh signal, refreshes the integration profiles.
5. On receipt of the shutdown signal, waits till all the connectors complete the synchronization / provisioning event propagation and returns to the main thread.

Connector Process Sequence

1. As part of initialisation, the connector establishes connection with Oracle Internet Directory and the connected directory. If the 'data interface type' is LDIF or Tagged then appropriate files are opened.
2. Reads the changes one at a time from the source.
3. Filter the changes if applicable.
4. Map the changes as specified by the mapping rules and create the destination change record.
5. Write the changes to the destination.
6. After applying all the changes, return back to the scheduler.

Managing Configuration Set Entries

As discussed above, a configuration set entry contains a list of all the integration profiles that the directory integration server is to execute. You can create, modify, and view configuration set entries by using either Oracle Directory Manager or the appropriate command line tools.

A configuration set is also a means of establishing an association between the host and the integration profile for synchronization. When a connector is registered, an integration profile is created and added to the configuration set. This configuration set entry determines the behavior of the directory integration server.

You can control the runtime behavior of the directory integration server by using a different configuration set entry when you start it. For example, you can start instance 1 of the directory integration server on host H1 with `configset1`, and instance 2 of the directory integration server on host H1 with `configset2`. The behavior of instance 1 of the directory integration server depends on configset 1, and that of instance 2 depends on configset2. By dividing different agents on host H1 between the two configuration set entries, you are distributing the load of running the agents on host H1 between the two directory integration server instances. Similarly, running different configuration sets and different instances on different hosts helps in balancing the load between the servers.

Managing the Oracle Directory Integration Server

This section contains these topics:

Starting the Oracle Directory Integration Server

The Oracle directory integration server executable, `odisrv`, resides in the $*ORACLE_HOME*`/bin` directory.

The way you start the directory integration server depends on whether your installation includes the **OID Monitor** and the **OID Control Utility**. These

tools—along with other server and client components—are parts of a typical server installation. In such installations, you start the directory integration server by using these tools.

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

Client-only installations do not include the OID Monitor and the OID Control Utility. In such installations, you start the directory integration server from the command line.

The directory server can be started in non-SSL mode, or in SSL mode for tighter security. Table 30–2 describes the parameters for each type of invocation.

> **Note:** When the Oracle directory integration server is invoked in the default mode, it supports only the Oracle Directory Provisioning Integration Service, and not the Oracle Directory Synchronization Service.

Using the OID Monitor and Control Utilities to Start the Oracle Directory Integration Server

To start the directory integration server in non-SSL mode:

1. Be sure that OID Monitor is running. To verify this, enter the following at the command line:

   ```
   ps -ef | grep oidmon
   ```

   If OID Monitor is not running, then start it by following the instructions in "Task 1: Start the OID Monitor" on page 3-2.

2. Start the directory integration server by using the OID Control utility by entering:

   ```
   oidctl [connect=net_service_name] server=odisrv [instance=instance_number]
   [config=configuration_set_number] [flags="[host=hostname] [port=port_number]
   [debug=debug_level] [refresh=interval-between-refresh]
   [maxprofiles=number-of-profiles] "] start
   ```

Table 30–2 describes the arguments in this command.

Table 30–2 Description of Arguments for Starting Oracle Directory Integration Server

Argument	Description
`connect=`*`net_service_name`*	If you already have a `tnsnames.ora` file configured, then this is the net service name specified in that file, located in $*`ORACLE_HOME`*`/network/admin`
`server=odisrv`	Type of server to start. In this case, the server you are starting is `odisrv`. This is not case-sensitive. This argument is mandatory.
`instance=`*`instance_number`*	Specifies the instance number to assign to the directory integration server. This instance number must be unique. OID Monitor verifies that the instance number is not already associated with a currently running instance of this server. If it is associated with a currently running instance, then OID Monitor returns an error message.
`config=`*`configuration_set_number`*	Specifies the number of the configuration set that the the directory integration server is to execute. This argument is mandatory.
`host=`*`hostname`*	Oracle directory server host name
`port=`*`port_number`*	Oracle directory server port number
`debug=`*`debug_level`*	The required debugging level of the directory integration server **See Also:** Table 30–4 on page 30-13 for a description of the various debug levels
refresh=*interval-between-refresh*	Specifies the interval, in minutes, between server refresh for any changes in the integration profiles. Default is 2 minutes (Refresh=2).
maxprofiles=*number-of-profiles*	Specifies the maximum number of profiles that can be executed concurrently for this server instance
sslauth=*ssl_mode*	SSL modes (0: NO Auth, 1: One Way)

Table 30–2 Description of Arguments for Starting Oracle Directory Integration Server

Argument	Description

wloc=*wallet*	SSL wallet. Enter the full path. For example, on Solaris, you could set this parameter as follows:
	`file:/home/my_dir/my_wallet`
	On Windows NT, you could set this parameter as follows:
	`file:C:\my_dir\my_wallet`

wpass=*wallet_password*	Password used for opening the SSL wallet

To start the directory server in SSL mode, use the following command:

```
oidctl [connect=net_service_name] server=odisrv [instance=instance_number]
[config=configuration_set_number] [flags="[host=hostname] [port=port_number]
[debug=debug_level] [refresh=interval-between-refresh] [maxprofiles=number-of-profiles]
[ sslauth=ssl_mode ] [ wloc=wallet ] [ wpass=wallet_password] "] start
```

As you can see, the only difference is the use of the SSL-related flags:
sslauth=*ssl_mode*, wloc=*wallet*, and wpass=*wallet_password*

Starting the Oracle Directory Integration Server Without Using OID Monitor and the OID Control Utility

The directory server can also be started without OID Monitor or OID Control Utility, either in non-SSL mode or, for tighter security, in SSL mode. The parameters described in Table 30–2 remain the parameters for each type of invocation.

To start the directory integration server in non-SSL mode, enter the following at the command line:

```
odisrv [host=host_name] [port=port_number]
config=configuration_set_number [instance=instance_number] [debug=debug_level]
[refresh=interval-between-refresh] [maxprofiles=number-of-profiles]
```

To start the directory integration server in SSL mode, enter the following at the command line:

```
odisrv [host=host_name] [port=port_number] config=configuration_set_number
[instance=instance_number] [debug=debug_level] [refresh=interval-between-refresh]
[maxprofiles=number-of-profiles] [ sslauth=ssl_mode ] [ wloc=wallet ]
[ wpass=wallet_password]
```

Again you can see that the only difference is the use of the SSL-related flags: sslauth=*ssl_mode*, wloc=*wallet*, and wpass=*wallet_password*

Stopping the Oracle Directory Integration Server

You stop the directory integration server using the same tool that you used to start it: by using OID Monitor and the OID Control Utility, or by using odisrv.

Using OID Monitor and the OID Control Utility to Stop the Server

If you started the directory integration server by using OID Monitor and the OID Control utility, then you use them to stop it, as follows:

1. Before you stop the directory integration server, be sure that the OID Monitor is running. To verify this, enter the following at the command line:

   ```
   ps -ef | grep oidmon
   ```

   If OID Monitor is not running, then start it by following the instructions in "Task 1: Start the OID Monitor" on page 3-2.

2. You then can stop the directory integration server by entering:

   ```
   oidctl [connect=net_service_name] server=odisrv instance=instance stop
   ```

Stopping the Directory Integration Server Without Using OID Monitor and the OID Control Utility

In a client-only installation where the monitor and OIDCTL tools are not available, Oracle directory integration server can be started without the OIDCTL tool. To stop the server without these tools, use the `stopOdiServer.sh` tool, which is located at
$*ORACLE_HOME*`/ldap/admin/stopodiserver.sh`.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

The `stopOdiServer.sh` tool is used as follows:

```
$ORACLE_HOME/ldap/admin/stopodiserver.sh  \
  [-LDAPhost <LDAP server host> ] [ -LDAPport <LDAP server port> ] \
  [-binddn SuperUserDN (default cn=orcladmin ) ]\
  [-bindpass Bindpassword (default=welcome) ]\
        -instance <Instance NUmber to STOP>
```

where the arguments are as explained in Table 30–3.

Table 30–3 Arguments for stopOdisServer Tool

Argument	Description
LDAPhost	The LDAP Server host. The default is the current host
LDAPport	The LDAP server port The default is port 389
Binddn	The BIND DN of the Directory user which has the privileges to create Integration profile. The default is 'cn=orcladmin'
Bindpass	The BIND password. The default is `welcome`
Instance	The instance number of the DIP server to stop

Note: If Oracle directory integration server is stopped by using any means other than the methods mentioned above, the server cannot be started from the same host. In that case, the footprint of the previous execution in the directory needs to be removed by the following command:

`$`*`ORACLE_HOME`*`/ldap/admin/stopodiserver.sh [-LDAPhost` *`LDAP_Server_Host`*`] [-LDAPhost` *`LDAP_Server_Port`*`] [ -binddn` *`Super_User_DN`* `(default is cn=orcladmin)] [ -bindpass` *`Super_User_Password`* `(default is welcome)] -instance` *`number_of_the_instance_to_stop`* **`-clean`**

Using the Restart Command

If you use OID Monitor and the OID Control utility, then you can both stop and restart the directory integration server in one command, namely, `restart`. This is useful when you want to refresh the server cache immediately, rather than at the next scheduled time. When the directory integration server restarts, it maintains the same parameters it had before it stopped.

To restart the directory integration server:

1. Make sure that OID Monitor is running. To verify this, enter the following at the command line:

   ```
   ps -ef | grep oidmon
   ```

   If OID Monitor is not running, then start it by following the instructions in "Task 1: Start the OID Monitor" on page 3-2.

2. At the command line, enter:

   ```
   oidctl [connect=net_service_name] server=odisrv instance=instance_number
   restart
   ```

Setting the Debug Level

You can specify the kinds of server and profile events to be listed in a log file by using the `debug` flag.

To specify multiple types of debugging:

1. Add the numeric values of the individual types as indicated in Table 30–4 on page 30-13.
2. At the command line, specify the total value. For example, the following command sets the debug level to 484:

   ```
   oidctl server=odisrv flags="debug=7" start
   ```

The various types of debug events are listed in Table 30–4 and Table .

Table 30–4 Debug Types for Server Debugging

Debug Event Type (Server debugging)	Numeric Value
Starting and stopping of different threads.	1
Detail level - shows the refresh details	2

Table 30–5 Debug Types for Profile Debugging

Debug Event Type (Profiles)	Numeric Value
Start and Stop of the thread	1

Table 30–5 (Cont.) Debug Types for Profile Debugging

Debug Event Type (Profiles)	Numeric Value
Initialization, execution, and end details	2
Details during execution	4
Change Record	8
Mapping Details	16

If you do not set a value for the debug flag, then the default level is `0` (zero), and none of the debug events in the tables above are logged. (However, errors and exceptions are *always* logged.)

When a non-zero debug level is specified, each trace statement in the server log file includes:

- Timestamp
- Thread type
- Profile name

The various trace-statement types are:

- `Main`

  Messages from the controller thread

- `Scheduler`

  Messages from the scheduler thread

Finding the Log Files

The log file is located in the $*ORACLE_HOME*/ldap/log/odisrv_*instance_number*.log directory.

For example, if the server was started as server instance number 3, then the log file would have this path name: $*ORACLE_HOME*/ldap/log/odisrv03.log.

All the profile-specific debug events are stored in the profile-specific trace file in $*ORACLE_HOME*\ldap\odi\log*profile_name.trc*.

Changing the Synchronization Status Attribute

While synchronization is in progress for an export operation, the server constantly updates the synchronization status attribute, `orcllastappliedchangenumber`. In Oracle Directory Manager, this field is called *OID last applied change number*.

To change this attribute manually from Oracle Directory Manager:

1. Disable the agent by using Oracle Directory Manager.
2. Make the attribute changes.
3. Re-enable the agent after the change.

Viewing Oracle Directory Integration Server Information

When the directory integration server starts, it generates specific runtime information and stores it in the directory. This information includes:

- Instance number of the directory integration server
- Host on which it is running
- Configuration set with which the directory integration server was started
- State of the configuration set refresh flag. This flag indicates to the directory integration server whenever there is a change to a directory integration profile and a refresh is required.

You can view this information for the directory integration server by using either Oracle Directory Manager or ldapsearch.

Viewing Oracle Directory Integration Server Runtime Information by Using Oracle Directory Manager

To view runtime information for the directory integration server instance by using Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory Servers > *directory_server_instance* > Server Management, then select Directory Integration Server. The Active Processes box appears in the right pane.
2. Click View Properties. The Server Process dialog box displays the information.

Viewing Oracle Directory Integration Server Runtime Information by Using ldapsearch

To view registration information for the directory integration server instance by using ldapsearch, perform a base search on its entry. For example:

```
ldapsearch -p 389 -h my_host -b cn=instance1,cn=odisrv,cn=subregistrysubentry -s
base -v "objectclass=*"
```

This example search returns the following:

```
dn: cn=instance1,cn=odisrv,cn=subregistrysubentry
cn: instance1
orcldiaconfigdns: "orclDIAName=HR,cn=subscriber profile,cn=changelog subscriber,
cn=oracle internet directory"
orcldiaconfigrefreshflag: 0
orclhostname: my_host
orclconfigsetnumber: 1
objectclass: top
objectclass: orclDIA
```

Managing the Oracle Directory Integration Platform in a Replicated Environment

If you use the Oracle Directory Integration Platform in a replicated environment with more than one node, then set the `orcldiprepository` attribute in DSE root to `1`. This makes the directory server generate change log entries for changes from the other Oracle Internet Directory nodes. (By default, the directory server does not generate these change log entries.) The change log entries are required for directory data to be synchronized with third-party directories and metadirectories.

31

Security in the Oracle Directory Integration Platform

This chapter discusses the most important aspects of security in the Oracle Directory Integration Platform. It contains these sections:

Authentication

Authentication is the process by which the Oracle directory server establishes the true identity of the user connecting to the directory. It occurs when an LDAP session is established by means of the ldapbind operation.

It is important that each component in the Oracle Directory Integration Platform be properly authenticated before it is allowed access to the directory.

Secure Sockets Layer (SSL) and the Oracle Directory Integration Platform

You can deploy the Oracle Directory Integration Platform either with or without **Secure Socket Layer (SSL)**. SSL implementation supports these modes:

- No authentication—Provides SSL encryption of data, but does not use SSL for authentication
- SSL server authentication—Includes both SSL encryption of data and SSL authentication of the server to the client. In the Oracle Directory Integration Platform, the server is the directory server, the client is the directory integration server.

  The server verifies its identity to the client by sending a **certificate** issued by a trusted **certificate authority (CA)**. This mode requires a public key infrastructure (PKI) and SSL wallets to hold the certificates.

To use SSL with the Oracle Directory Integration Platform, you must start both the Oracle directory server and Oracle directory integration server in the SSL mode.

> **See Also:** Chapter 3, "Preliminary Tasks and Information" for instructions on starting the Oracle directory server in SSL mode

Oracle Directory Integration Server Authentication

You can install and run multiple instances of the directory integration server on various hosts. When you do this, beware of a malicious user either posing as the directory integration server or using an unauthorized copy of it.

To avoid such security issues:

- Ensure that each directory integration server is identified properly
- Ensure that, when you start a directory integration server, it is properly authenticated before it obtains access to Oracle Internet Directory

Non-SSL Authentication

To use non-SSL authentication, register each directory integration server by using the registration tool called odisrvreg.

The registration tool creates:

- An identity entry in the directory. The directory integration server uses this entry when it binds to the directory
- An encrypted password. It stores this password in the directory integration server entry.
- A private wallet on the local host. This wallet contains the security credentials, including an encrypted password. The name of the wallet is `odisrvwallet`, and it is stored in the `$`*`ORACLE_HOME`*`/ldap/odi/conf` directory.

When it binds to the directory, the directory integration server uses the encrypted password in the private wallet.

Note: Ensure that the wallet is protected against unauthorized access.

See Also: "Registering the Oracle Directory Integration Server" on page 30-2 for instructions about registering the directory integration server

Authentication in SSL Mode

The identity of the directory server can be established by starting both Oracle Internet Directory and the directory integration server in the SSL server authentication mode. The directory server provides its certificate to the directory integration server, which acts the client of Oracle Internet Directory.

The directory integration server is authenticated by using the same mechanism used in the non-SSL mode.

Profile Authentication

Within Oracle Internet Directory, an integration profile represent a user with its own DN and password. This information is stored in the integration profile of the agent. To protect the profile from unauthorized access, establish appropriate access control policies for it in the directory such that only an integration platform

administrator or a user designated by the Oracle Internet Directory administrator can create the integration profiles.

When the directory integration server imports data to OID based on an integration profile, it binds to the directory as that integration profile and uses the profile name and password for binding. The Oracle Directory Integration Platform uses this mechanism to authenticate agents in both the SSL and non-SSL mode.

Access Control and Authorization

Authorization is the process of ensuring that a user reads or updates only the information for which that user has privileges. When directory operations are attempted within a directory session, the directory server ensures that the user—identified by the authorization identifier associated with the session—has the requisite permissions to perform those operations. Otherwise, the operation is disallowed. Through this mechanism, the directory server protects directory data from unauthorized operations by directory users. This mechanism is called access control.

To restrict access to only the desired subset of Oracle Internet Directory data, for both the directory integration server and the agents, place appropriate access policies in the directory. The following section discusses these policies in detail.

Access Controls for the Oracle Directory Integration Server

The directory integration server binds to the directory both as itself and on behalf of the agent.

- When it binds as itself, it can cache the information in various integration profiles. This enables the directory integration server to schedule synchronization actions to be carried out by various connectors.
- When the directory integration server operates on behalf of an agent, it uses the agent credentials to bind to the directory and perform various operations. The directory integration server can perform only those operations in the directory that are permitted to the agent.

To establish and manage access rights granted to directory integration servers, the Oracle Directory Integration Platform creates a group entry, called `odisgroup`, during installation. When a directory integration server is registered, it becomes a member of this group.

You control the access rights granted to directory integration servers by placing access control policies in the `odisgroup` entry. The default policy grants various

rights to directory integration servers for accessing the profiles. For example, the default policy enables the directory integration server to compare user passwords for authenticating agents when it binds on their behalf. It also enables directory integration servers to modify status information in the profile—such as the last successful execution time and the synchronization status.

Access Controls for Agents

To control access to Oracle Internet Directory data by integration profiles, place appropriate access control policies in Oracle Internet Directory. This enables you to protect data synchronized or processed by one agent from interference by other agents. It also enables you to allow only the integration profile that owns synchronization of an attribute to modify that attribute.

See Also:

- "Adding Group Entries by Using Oracle Directory Manager" on page 7-8
- "Access Control Groups" on page 13-3 for instructions on setting access control policies for group entries.

For example, creating a group entry called `odipgroup` when installing the Oracle Internet Directory enables controlling the access rights granted to various agents. Rights are controlled by placing appropriate access policies in the `odipgroup` entry. Each agent is a member of this group. The membership is established when the agent is registered in the system. The default access policy, automatically installed with the product, grants to agents certain standard access rights for the integration profiles they own. One such right is the ability to modify status information in the integration profile, such as the parameter named `orclodipConDirLastAppliedChgTime`. The default access policy also permits agents to access Oracle Internet Directory change logs, to which access is otherwise restricted.

The `odisgroup` group entries and their default policies are created during the server installation of the Oracle Internet Directory. Client-only installations do not create these groups and policies.

Data Integrity

The Oracle Directory Integration Platform ensures that data has not been modified, deleted, or replayed during transmission by using SSL. This SSL feature generates a

cryptographically secure message digest—through cryptographic checksums using either the MD5 algorithm or the Secure Hash Algorithm (SHA) —and includes it with each packet sent across the network.

Data Privacy

The Oracle Directory Integration Platform ensures that data is not disclosed during transmission by using public-key encryption available with SSL. In public-key encryption, the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the recipient decrypts the message using the recipient's private key.

To exchange data securely between the directory integration server and Oracle Internet Directory, you run both components in the SSL mode.

Tools Security

You can run all the commonly used tools in the SSL mode to transmit data to Oracle Internet Directory securely. These tools include:

- Oracle Directory Manager —Use it to administer data in the directory
- The Oracle directory integration server registration tool (odisrvreg)—Use it to register the directory integration server in the directory
- Ldapadd and ldapmodify tools—Use these to add or modify entries from the command line

32

Bootstrapping of a Directory in the Oracle Directory Integration Platform

This chapter contains these topics:

> **Note:** The bootstrapping procedures in this chapter assume that the integration profile for a connected directory is available to synchronize between the connected directory and Oracle Internet Directory. The procedures are for only the initial synchronization or migration of data from one directory to the other.

Bootstrapping Oracle Internet Directory from a Connected Directory

When the connected directory is the source of truth, bootstrapping involves migration of data from the connected directory to Oracle Internet Directory. This can be achieved using any of the following methods:

- Using External Tools to Import Data into Oracle Internet Directory
- Setting up a Connector to Import Data in Oracle Internet Directory

If the directory from which the Oracle Internet Directory is being bootstrapped is also going to be part of the Oracle Directory Integration Platform environment—as is the case, for example, with Oracle HR—then follow the steps in this chapter for the initial bootstrap.

Using External Tools to Import Data into Oracle Internet Directory

1. Disallow any updates to the connected directory by setting it to read-only mode.
2. Dump the data from the connected directory in an LDIF file format or in an LDIF template format.
3. If the data is in the LDIF format, use the bulkload tool to upload the data to Oracle Internet Directory.
4. After copying is complete and verified, set the connected directory back to the update mode.

See Also: "Command-Line Tools Syntax" on page A-4 for instructions on using command-line tools

Setting up a Connector to Import Data in Oracle Internet Directory

With this method, the connector pulls changes from the connected directory based on either a timestamp (`orclOdipLastSuccessfulExecutionTime` attribute) or 'Last Applied Change Number' (`orclOdipConDirLastAppliedChgNum`) depending on how the connected directory identifies its changes. The data can be bootstrapped onto OID in the following sequence:

1. Disallow any updates to the connected directory by setting it to the read-only mode.
2. Register the connected directory agent with the Oracle Directory Integration Platform by using the Oracle Directory Manager.

See Also: "Registering the Oracle Directory Integration Server" on page 30-2

3. The Directory Integration Server starts the import operation based on the scheduling interval. Wait for the synchronization to complete.
4. Once the synchronization is complete and verified, set the connected directory back to the update mode.

Bootstrapping a Connected Directory from Oracle Internet Directory

Similar to bootstrapping OID from a connected directory, bootstrapping the connected directory from OID can also be achieved by either of the methods described in these topics:

- Using External Tools to Export Data from OID
- Setting up a Connector to Export Data from OID.

Using External Tools to Export Data from OID

1. Disallow any updates to Oracle Internet Directory by setting it to the read-only mode.
2. Dump the data from OID to an LDIF file using the LDIFWrite tool.
3. Use the LDIF file to load data to the connected directory.

Setting up a Connector to Export Data from OID

With this method, the changes from OID are pushed to the connected directory based on OID Last applied change number (orclLastAppliedChangeNumber attribute). The data can be bootstrapped onto the connected directory in the following sequence:

1. Disallow any updates to Oracle Internet Directory by setting it to the read-only mode.
2. Register the connected directory agent with the Oracle Directory Integration Platform using the Oracle Directory Manager.
3. The Directory Integration Server starts the export operation based on the scheduling interval. Wait for the export to complete.

4. Once the synchronization is complete and verified, set the connected directory back to the update mode.

33

Synchronization with Oracle Human Resources

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

This chapter introduces the Oracle Human Resources connector and explains how to deploy it. It contains these topics:

Introduction

The Oracle Human Resources connector enables you to import a subset of employee data from Oracle Human Resources into Oracle Internet Directory. It is installed, with a default configuration, along with Oracle Internet Directory. It is ready to run out of the box.

You can schedule the Oracle Human Resources connector to run at any time, configuring it to extract incremental changes from the Oracle Human Resources system as often as every second. You can also set and modify attribute mapping between Oracle Human Resources and Oracle Internet Directory.

The Oracle Human Resources connector executable name is `odihragent` and is located in the `$ORACLE_HOME/ldap/odi/bin` directory. You can manage the Oracle Human Resources connector by using Oracle Directory Manager.

Data that You Can Import from Oracle Human Resources

Table 33–1 lists the tables in the Oracle Human Resources schema, most of whose attributes you can import into Oracle Internet Directory:

Table 33–1 Tables in Oracle Human Resources Schema

Table Name	Alias Used in the Connector Config Info Field
PER_PEOPLE_F	PER
PER_ADDRESSES	PA
PER_PERIOD_OF_ SERVICE	PPS
PER_PERSON_TYPE	PPT

All of these tables are visible if the login to the Oracle Human Resources database is done with the apps account.

Because attributes can be added or deleted at runtime from the configuration file, the Oracle Human Resources connector dynamically creates a SQL statement that selects and retrieves only the required attributes.

Table 33–2 shows some of the fields in the Oracle Human Resources user interface. These fields appear when you add or modify employee data.

Table 33–2 Fields in the Oracle Human Resources User Interface

ATTRIBUTE NAME	DESCRIPTION	FORM/CANVAS/FIELD_NAME
LAST_NAME	Last name of the person	People / Name / Last
FIRST_NAME	First name of the person	People / Name / First
TITLE	Title of the person	People / Name / Title
SUFFIX	Suffix—for example, Jr, Sr, Ph.D.	People / Name / Suffix
MIDDLE_NAME	Middle name	People / Name / Suffix
SEX	Sex	Gender List box
START_DATE	Hiring date	People / Hire Date
DATE_OF_BIRTH	Date of birth	People / Personal Information / Birth Date
MARITAL_STATUS	Marital status	People / Personal Information / Status
NATIONAL_ INDENTIFIER	Social security number for US residents	People / Identification / Social Security
EMPLOYEE_NUMBER	Employee number	People / Identification / Employee
REGISTERD_ DISABLED_ FLAG	Indicator that the employee has a disability	People / Personal Information / Has Disability
EMAIL_ADDRESS	Electronic mail address	People / Personal Information / EMail
OFFICE_NUMBER	Office location	People / Office Location Info / Office
MAILSTOP	Mail delivery stop	People / Office Location Info / Mail Stop
INTERNAL_ LOCATION	Location	People / Office Location Info / Location
ADDRESS_LINE1		Personal Address Information / Address line 1
ADDRESS_LINE2		Personal Address Information / Address line 2
ADDRESS_LINE3		Personal Address Information / Address line 3
TOWN_OR_CITY		Personal Address Information / City
REGION_1		Personal Address Information / County

Table 33–2 Fields in the Oracle Human Resources User Interface

ATTRIBUTE NAME	DESCRIPTION	FORM/CANVAS/FIELD_NAME
REGION_2		Personal Address Information / State
POSTAL_CODE		Personal Address Information / Zip Code
COUNTRY		Personal Address Information / Country
TELEPHONE_ NUMBER_1		Personal Address Information / Telephone
TELEPHONE_ NUMBER_2		Personal Address Information / Telephone2

Managing Synchronization with Oracle Human Resources

This section contains these topics:

- Configuring a Directory Integration Profile for the Oracle Human Resources Connector
- Customizing the List of Attributes to Be Synchronized with Oracle Internet Directory
- Customizing Mapping Rules for the Oracle Human Resources Connector
- Running Synchronization from Oracle Human Resources to Oracle Internet Directory

Configuring a Directory Integration Profile for the Oracle Human Resources Connector

To deploy the Oracle Human Resources connector, you must create a directory integration profile for it in Oracle Internet Directory. You can do this by using the procedures outlined in Chapter 30, "Oracle Directory Integration Server Administration". However, if you have a server installation—that is, a typical installation—then you can use the default integration profile that the Oracle Universal Installer created in the directory for you. A client-only installation does not include this integration profile.

The integration profile contains several attributes and attribute values. Table 33–3 lists these attributes by both their friendly names as used by Oracle Directory Manager—for example, `Agent Name`—and their actual names—for example, `orclodipAgentName`. It provides a description of each attribute, and, where appropriate, the default values in the Oracle Human Resources connector

integration profile. Some cells in Table 33–3 contain italicized text providing information and instructions specific to the Oracle Human Resources connector

Table 33–3 Attributes in the Oracle Human Resources Connector Integration Profile

Attribute	Description
General Information	
Profile Name (`orclODIPAgentName`)	Unique name by which the connector is identified in the system, used as an RDN component of the DN that identifies the integration profile. The name can contain only alpha-numeric characters. This attribute is mandatory and not modifiable. The default name is OracleHRAgent.
Profile Status (`orclODIPAgentcontrol`)	Indicates whether the connector is enabled or disabled. Valid values are ENABLE or DISABLE. The default is DISABLE. This attribute is mandatory and modifiable. You must set this value to ENABLE.
Profile Password (`orclODIPAgentPassword`)	This is the password that the directory integration server uses to bind to Oracle Internet Directory on behalf of the profile. This attribute is mandatory and modifiable. Set this value to whatever password you want the Oracle Human Resources Profile to use.
Execution Host (`orclODIPAgentHostName`)	Host on which the connector runs. This attribute is mandatory and modifiable. This attribute is currently ignored.
Synchronization Mode (`orclODIPSynchronizationMode`)	The direction of synchronization between Oracle Internet Directory and a connected directory. IMPORT indicates importing changes from a connected directory to Oracle Internet Directory. EXPORT indicates exporting changes from Oracle Internet Directory to a connected directory. The default is IMPORT. This attribute is mandatory and modifiable. Note: Oracle Internet Directory release 9.0.2 support import operations only, for the Oracle HR.
Scheduling Interval (`orclODIPSchedulingInterval`)	Time interval in seconds after which a connected directory is synchronized with Oracle Internet Directory. The default is 600. This attribute is mandatory and modifiable.

Table 33–3 Attributes in the Oracle Human Resources Connector Integration Profile

Attribute	Description
Max Number of Retries (`orclODIPSyncRetryCount`)	Maximum number of times the directory integration server would try to perform synchronization before giving up. It will then retry at the next scheduled time. The default is 5. This attribute is mandatory and modifiable.
Execution Information	
Agent Execution Command (`orclODIPAgentExeCommand`)	Connector executable name and argument list used by the directory integration server to execute the connector. This attribute is mandatory and modifiable. The default is: odihragent connect=hrdb \ login=%orclodipConDirAccessAccount \ pass=%orclodipConDirAccessPassword \ date=%orclODIPLastSuccessfulExecutionTime \ You must set the value in the argument connect=hrdb to the connect string of the Oracle Human Resources system database.
Connected Directory Account (`orclODIPConDirAccessAccount`)	Valid user account in the Oracle Human Resources system that you want to access changes in the Oracle Human Resources system. This information is passed by the directory integration server to the connector in the command line at time of connector's invocation. This attribute is optional and modifiable.
Connected Directory Account Password (`orclODIPConDirAccessPassword`)	Password for the user account accessing the Oracle Human Resources system. It is passed by the Directory integration server to the connector at time of connector invocation. This attribute is optional and modifiable.

Table 33–3 Attributes in the Oracle Human Resources Connector Integration Profile

Attribute	Description
Additional Config Info (`orclODIPAgentConfigInfo`)	Any configuration information that you want an connector to store in Oracle Internet Directory. It is passed by the directory integration server to the connector at time of connector invocation. The information is stored as an attribute and the directory integration server does not have any knowledge of its content. The value stored in this attribute represents (for Oracle Human Resources connector) all attributes that need to be synchronized from Oracle Human Resources. It is discussed in "Customizing the List of Attributes to Be Synchronized with Oracle Internet Directory". This attribute is mandatory for Oracle Human Resources connector, and modifiable by editing the configuration file and uploading it again into the profile.
Interface Type (`orclODIPInterfaceType`)	The Interface being used for data transfer. Since it is in the form of a TAGGED file, it is set to 'TAGGED'. Note: You should not modify this attribute for Oracle HR Profile.
Mapping Information	
Mapping Rules (`orclODIPAttributeMappingRules`)	The mapping rules for mapping data between a connected directory and Oracle Internet Directory. The value stored in this attribute is discussed under "Mapping Rules". This attribute is mandatory for Oracle HR and modifiable.
Connected Directory Matching Filter (`orclODIPConDirMatchingFilter`)	This is not used in Oracle HR Connectivity.
OID Matching Filter (`orclODIPOIDMatchingFilter`)	This attribute names an LDAP filter that is used to search for a target entry in OID. The Server uses this filter to find out what kind of LDAP operation it needs to do to synchronize. It is of the form "employeenumber=%" It is optional and modifiable.
Status Information	
Synchronization Status (`orclODIPSynchronizationStatus`)	Indicates the execution status of the profile as it is synchronizing. This is read-only.

Table 33–3 Attributes in the Oracle Human Resources Connector Integration Profile

Attribute	Description
Synchronization Errors (`orclODIPSynchronizationErrors`)	Error message for the last error encountered in synchronization. This is read-only.
Last Execution Time (`orclODIPLastExecutionTime`)	Time of the most recent profile execution. It is generally read-only. It can be modified to re-sync from a different point in time.
Last Successful Execution Time (`orlcODIPLastSuccessfulExecutionTime`)	Time of the most recent successful profile execution. It is generally read-only. It can be modified to re-sync from a different point in time.
Connected Directory Last Applied Change Number (`orclODIPConDirLastAppliedChgNum`)	This attribute, standard for all profiles, does not apply to the Oracle HR Synchronization.
OID Last Applied Change Number (`orcllastappliedChangenumber`)	This attribute, standard for all EXPORT profiles, does not apply to Oracle HR sync.

Customizing the List of Attributes to Be Synchronized with Oracle Internet Directory

You can customize the list of Oracle Human Resources attributes you want to synchronize with Oracle Internet Directory. To help you do this, Oracle Internet Directory includes a default list of Oracle Human Resources attributes to be synchronized. You can modify this list by including additional attributes in it, or excluding some from it.

The default attribute list is stored in the `orclodipAgentConfigInfo` attribute as part of the integration profile. The integration profile is loaded into Oracle Internet Directory as part of a typical installation. The list is also contained in the file named `oraclehragent.cfg.master` and is located under the $*ORACLE_HOME*`/ldap/odi/conf` directory.

Note: Do not modify the `oraclehragent.cfg.master` file; it serves as a backup.

The columns in the default list of Oracle Human Resources attributes are:

Column	Description
ATTRNAME	The output tag generated in the output data file
COLUMN_NAME	Database column name from where to obtain this value
TABLE_NAME	Database table name from where to obtain this value
FORMAT	The column data type of this attribute. (ASCII, NUMBER, DATE)
MAP	Indicator of whether to extract this attribute from Oracle Human Resources or not. A value of `Y` indicates that it will be extracted and a value of `N` indicates that it will not be.

The `oraclehragent.cfg.master` file contains the following:

```
ATTRNAME:COLUMN_NAME:TABLE_NAME:FORMAT:MAP
PersonId:person_id:PER:NUMBER:Y
PersonType:person_type_id:PER:NUMBER:Y
PersonTypeName:system_person_type:PPT:ASCII:Y
LastName:last_name:PER:ASCII:Y
StartDate:start_date:PER:DATE:Y
BirthDate:date_of_birth:PER:DATE:Y
EMail:email_address:PER:ASCII:Y
EmployeeNumber:employee_number:PER:NUMBER:Y
FirstName:first_name:PER:ASCII:Y
FullName:full_name:PER:ASCII:Y
knownas:known_as:PER:ASCII:Y
MaritalStatus:marital_status:PER:ASCII:Y
middleName:middle_names:PER:ASCII:Y
country:country:PA:ASCII:Y
socialsecurity:national_identifier:PER:ASCII:Y
Sex:sex:PER:ASCII:Y
Title:title:PER:ASCII:Y
suffix:suffix:PER:ASCII:Y
street1:address_line1:PA:ASCII:Y
zip:postal_code:PA:ASCII:Y
Address1:address_line1:PA:ASCII:Y
Address2:address_line2:PA:ASCII:Y
Address3:address_line3:PA:ASCII:Y
TelephoneNumber1:telephone_number_1:PA:ASCII:Y
TelephoneNumber2:telephone_number_2:PA:ASCII:Y
TelephoneNumber3:telephone_number_3:PA:ASCII:Y
town_or_city:town_or_city:PA:ASCII:Y
```

```
state:region_2:PA:ASCII:Y
Start_date:effective_start_date:PER:DATE:Y
End_date:effective_end_date:PER:DATE:Y
per_updateTime:last_update_date:PER:DATE:Y
pa_updateTime:last_update_date:PA:DATE:Y
```

Including Additional Oracle Human Resources Attributes for Synchronization

To include additional Oracle Human Resources attributes for synchronization, follow these steps:

1. Copy the `oraclehragent.cfg.master` file and name it anything other than *Agent_Name*.cfg. This is because the directory integration server generates a configuration file with that name, using it to pass the configuration information to the Oracle Human Resources connector at run time.
2. Include an additional Oracle Human Resources attribute for synchronization by adding a record to this file. To do this, you need this information:
   - Table name in the database from which the attribute value is to be extracted. These tables are listed in Table 33–1 on page 33-2. The file uses abbreviated names for the four tables used in the synchronization.
   - Column name in the table
   - Column datatype. Valid values are ASCII, NUMBER, DATE

   You also need to assign an attribute name to the column name. This acts as the output tag by which this attribute is identified in the output file. This tag is also used in the mapping rules to establish a rule between the Oracle Human Resources attribute and the Oracle Internet Directory attribute.

   You must also ensure that the `map` column—that is, the last column in the record—is set to the value Y.

Note: If you add a new attribute in the attribute list, then you must define a corresponding rule in the `orclodipAttributeMappingRules` attribute. Otherwise the Oracle Human Resources attribute is not synchronized with the Oracle Internet Directory even if it is being extracted by the Oracle Human Resources connector. See "Creating Oracle Human Resources Attribute Mapping Rules" on page 33-14 for instructions about creating mapping rules.

---

3. Load the file into the `orclodipAgentConfigInfo` attribute by using the ldapmodify tool. The changes take effect the next time the connector runs.

Excluding Oracle Human Resources Attributes from Synchronization

To exclude an Oracle Human Resources attribute that is currently being synchronized with Oracle Internet Directory:

1. Copy the `oraclehragent.cfg.master` file and name it anything other than *Agent_Name*.cfg. This is because the directory integration server generates a configuration file with that name, using it to pass the configuration information to the Oracle Human Resources connector at run time.
2. Do one of the following:
   - Comment out the corresponding record in the attribute list by putting a hash sign (#) in front of it
   - Set the value of the column `map` to `N`
3. Load the file into the `orclodipAgentConfigInfo` attribute by using the ldapmodify tool. The changes take effect the next time the connector runs.

Configuring a SQL SELECT Statement in the Configuration File to Support Complex Selection Criteria

If the above supporting Attribute Configuration is not sufficient to extract data from the HR Database, then the Oracle Human Resources connector also supports execution of a pre-configured SQL SELECT statement in the configuration file. There is a TAG to indicate this in the config file, namely, a “[SELECT]” in the config file.

The following example shows a sample select statement to fetch some information from the HR database. Note that only the SQL statement should be below the [SELECT] Tag. The BINDVAR Bind Variable needs to be there to fetch incremental changes. The substitutes passes this value (the time stamp) to the Oracle Human Resources connector.

All the columns expressions fetched in the SELECT must have column names—for example, REPLACE(ppx.email_address),'@ORACLE.COM','') is fetched as EMAILADDRESS. The Oracle Human Resources connector writes out EMAILADDRESS as the attribute name in the output file with its value as the result of the expression REPLACE(ppx.email_address),'@ORACLE.COM'".

```
[SELECT]

SELECT
     REPLACE(ppx.email_address),'@ORACLE.COM',''), EMAILADDRESS ,
     UPPER(ppx.attribute26) GUID,
     UPPER(ppx.last_name) LASTNAME,
     UPPER(ppx.first_name) FIRSTNAME,
     UPPER(ppx.middle_names) MIDDLENAME,
     UPPER(ppx.known_as) NICKNAME,
     UPPER(SUBSTR(ppx.date_of_birth,1,6)) BIRTHDAY,
     UPPER(ppx.employee_number) EMPLOYEEID,
     UPPER(ppos.date_start)  HIREDATE,
FROM
   hr_organization_units hou,
   per_people_x ppx,
   per_people_x mppx,
   per_periods_of_service ppos
  WHERE
      pax.supervisor_id  = mppx.person_id(+)
  AND pax.organization_id = hou.organization_id(+)
  AND ppx.person_id = ppos.person_id
  AND ppx.person_id = pax.person_id
  AND ppos.actual_termination_date IS NULL
  AND UPPER(ppx.current_employee_flag) = 'Y'
  AND ppx.last_update_date >= (:BINDVAR,'YYYYMMDDHH24MISS')
```

Customizing Mapping Rules for the Oracle Human Resources Connector

Attribute mapping rules govern how the directory integration server converts attributes between Oracle Human Resources and Oracle Internet Directory. You can customize the mapping rules you want the directory integration server to use.

To help you do this, Oracle Internet Directory includes a default list of Oracle Human Resources mapping rules for the Oracle Human Resources system. You configure, modify, and delete mapping rules by editing this list.

The default list of mapping rules is stored in the `orclodipAttributeMappingRules` attribute in the integration profile. In addition, the rules are also in the file named `oraclehragent.map.master` located under the $*ORACLE_HOME*`/ldap/odi/conf` directory.

Note: Do not modify the `oraclehragent.map.master` file; it serves as a backup.

Default Oracle Human Resources Connector Mapping Rules

The oraclehragent.map.master file contains the following:

```
DomainRules
DomainRules
NONLDAP:dc=metaagt,dc=com:uid=%dc=metaagt,dc=com
AttributeRules
firstname: : : :cn: :person
email : : : :cn: :person: trunc(email,'@')
email : : : :uid: :person:trunc(email,'@')
firstname,lastname: : : :cn: :person: firstname+","+lastname
lastname,firstname: : : :cn: :person: lastname+","+firstname
firstname,lastname: : : :sn: :person: lastname | firstname
EmployeeNumber: : : :employeenumber: :inetOrgperson
EMail: : : :mail: :inetOrgperson
TelephoneNumber1: : : :telephonenumber: :person
TelephoneNumber2: : : :telephonenumber: :person
TelephoneNumber3: : : :telephonenumber: :person
Address1: : : :postaladdress: :person
state: : : :st: :locality
street1: : : :street: :locality
zip: : : :postalcode: :locality
town_or_city: : : :l: :locality
Title: : : :title: :organizationalperson
#Sex: : : :sex: :person
```

In the sample mapping file, 'dc=metaagt,dc=com' is assumed as the 'synchronization domain'. This domain name needs to be changed according to deployment requirements.

```
AttributeRules
firstname: : : :cn: :person
lastname: : : :sn: :person
lastname: : : :cn: :person
 : : : :cn: :person: trunc(email,'@')
 : : : :cn: :person: firstname+","+lastname
 : : : :cn: :person: lastname+","+firstname
EmployeeNumber: : : :employeenumber: :inetOrgperson
EMail: : : :mail: :inetOrgperson
TelephoneNumber1: : : :telephonenumber: :person
TelephoneNumber2: : : :telephonenumber: :person
TelephoneNumber3: : : :telephonenumber: :person
Address1: : : :postaladdress: :person
Address1: : : :postaladdress: :person
Address1: : : :postaladdress: :person
```

```
state: : : :st: :locality
street1: : : :street: :locality
zip: : : :postalcode: :locality
town_or_city: : : :l: :locality
Title: : : :title: :organizationalperson
Sex: : : :sex: :person
socialsecurity: : : :ssn: :person
country: : : :c: :country
BirthDate: : : :birthday: :organizationalperson
 : : : :userpassword: :person: "welcome"
changetype
###
```

The default mapping rules in the `orclodipAttributeMappingRules` attribute correspond to the default Oracle Human Resources attributes list in the `orclodipAgentConfigInfo` attribute. To establish mappings between Oracle Human Resources attributes and Oracle Internet Directory attributes, the mapping rules use the ATTRNAME column in each record of the Oracle Human Resources attributes list.

> **See Also:** "Mapping Rules and Formats" on page 29-9 for the description of the format of the mapping rules records

Creating Oracle Human Resources Attribute Mapping Rules

To create Oracle Human Resources attribute mapping rules, you modify the `orclodipAttributeMappingRules` attribute. To do this:

1. Copy the `oraclehragent.map.master` file to *`Agent_Name`*`.map`.
2. Add a new rule to this file by adding a record to it. To do this, you need this information:
   - The Oracle Human Resources attribute name that is mapped to Oracle Internet Directory
   - The corresponding attribute in Oracle Internet Directory and its object class to which the Oracle Human Resources attribute are to map
   - The import rule that determines how to map the Oracle Human Resources attribute to the Oracle Internet Directory attribute
3. Load the file into the `orclodipAttributeMappingRules` attribute by using the following script: $*`ORACLE_HOME`*`/ldap/odi/admin/ldapUploadAgentFile.sh`. The changes take effect the next time the profile runs.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Modifying Oracle Human Resources Attribute Mapping Rules

To modify existing Oracle Human Resources attribute mapping rules, you modify the `orclodipAttributeMappingRules` attribute. To do this:

1. Copy the `oraclehragent.map.master` file *`Agent_Name`*`.map`.
2. Edit this file.
3. Load the file into the `orclodipAttributeMappingRules` attribute by using the ldapmodify tool. The changes take effect the next time the connector runs.

Deleting Oracle Human Resources Attribute Mapping Rules

To delete existing Oracle Human Resources attribute mapping rules, you modify the `orclodipAttributeMappingRules` attribute. To do this:

1. Copy the `oraclehragent.map.master` file and name it anything other than *`Agent_Name`*`.map`, which is reserved for use by the directory integration server.
2. Do one of the following:
   - Delete the rule from the file
   - Comment it out by putting a hash (#) sign in front of it.
3. Load the file into the `orclodipAttributeMappingRules` attribute by using the following script: $*`ORACLE_HOME`*`/ldap/odi/admin/ldapUploadAgentFile.sh`. The changes take effect the next time the profile runs.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Running Synchronization from Oracle Human Resources to Oracle Internet Directory

This section explains how to set up synchronization from Oracle Human Resources to Oracle Internet Directory.

During synchronization, the Oracle Directory Integration Platform uses an import file. This file can contain a few or many changes that the Oracle Human Resources connector extracts from the Oracle Human Resources system.

This file is in the tagged format and acts as input to the Oracle directory server. It is named *Oracle_HR_Agent_Name*.data and is located in $*ORACLE_HOME*/ldap/odi/import.

You do not need to modify this file, but the last version of it is stored in the directory $*ORACLE_HOME*/ldap/odi/import/archive to help you with troubleshooting.

This is an example of an Oracle Human Resources change record in the import file:

```
FirstName: John
LastName: Liu
EmployeeNumber: 12345
Title: Mr.
Sex: M
MaritalStatus: Married
TelephoneNumber: 123-456-7891
Mail: Jliu@my_company.com
Address: 100 Jones Parkway
City: MyTown
```

Preparing for Synchronization

To prepare for synchronization between Oracle Human Resources and Oracle Internet Directory, follow these steps:

1. Ensure that the Oracle Human Resources connector and the directory integration server are installed on the host from which you want to run the Oracle Human Resources connector.

   See Also: The file `install.txt` and the Readme file for Oracle Internet Directory Release 9.0.2 for more details

2. Ensure that you have the information for accessing the Oracle Human Resources system, including:
   - Connect string to the Oracle Human Resources system database
   - Access account
   - Password
3. Ensure that the directory integration server on this host is registered in Oracle Internet Directory.

   See Also: "'Registering the Oracle Directory Integration Server" on page 30-2 for registration instructions

4. Configure an integration profile for the Oracle Human Resources connector, as described in "Configuring a Directory Integration Profile for the Oracle Human Resources Connector" on page 33-4. Ensure that all values in the integration profile are properly set, including:
   - Oracle Human Resources attribute list
   - Oracle Human Resources attribute mapping rules
   - Scheduling interval
5. Once everything is properly set, set the `orclodipAgentControl` attribute to `ENABLE`. This indicates that the Oracle Human Resources connector is ready to run.
6. Start the Oracle directory server and the Oracle Human Resources system if they are not already running on the respective hosts.
7. When everything is ready, start the directory integration server if it is not already running on this host.

> **See Also:** "Managing the Oracle Directory Integration Server" on page 30-7 for instructions about starting and stopping the directory integration server

The Synchronization Process

Once the Oracle Human Resources system, Oracle Internet Directory, and the directory integration server are running and the Oracle Human Resources connector is enabled, the directory integration server automatically starts synchronizing changes from the Oracle Human Resources system into Oracle Internet Directory. It follows this process:

1. Depending on the value specified in the 'Last Execution Time' (orclodipLastExecutionTime) and the 'scheduling interval' (orclodipschedulinginterval) the directory integration server invokes the 'Oracle Human Resources Connector'.
2. The Human Resources Connector extracts all the changes from the Oracle Human Resources System based on the time specified in the `orclodipLastSuccessfulExecutionTime` attribute in the integration profile. It writes the changes into the Oracle Human Resources Import file, namely `$`*`ORACLE_HOME`*`/ldap/odi/import/HR_Agent_Name.dat`. It extracts only the attributes specified in the orclodipAgentConfigInfo attribute in the Integration Profile.
3. After the agent completes the execution, the directory integration server imports the changes to Oracle Internet Directory by doing the following:
   - It reads each change record from the import file.
   - It converts each change record into and LDAP change entry based on the rules specified in the 'Mapping Rules' (orclodipAttributeMappingRules) in the Integration Profile.
4. After successfully importing all the changes successfully to Oracle Internet Directory, Oracle Human Resources Connector moves the import file to the archive directory, $ORACLE_HOME/ldap/odi/import/archive. The Status attributes Last Execution Time' (orclodipLastExecutionTime) and 'Last Successful Execution Time' orclodipLastSuccessfulExecutionTime attributes to the current time.
5. If the import operation fails, only the 'Last Execution Time' is updated, and the connector will once again attempt to extract the changes from Human Resources System based on the 'Last Successful Execution Time'. The reason for

failure will be logged in the trace file in $*ORACLE_HOME*/ldap/odi/HR_Agent_Name.trc file.

Boostrapping Oracle Internet Directory from Oracle HR

There are two ways to bootstrap Oracle Internet Directory from Oracle HR:

- Use the Oracle Human Resources connector. In the integration profile, set the orclodipConDirLastAppliedChgTime to a time before Oracle Human Resources was installed.
- Use external tools to migrate data from Oracle Human Resources into Oracle Internet Directory

See Also: Chapter 32, "Bootstrapping of a Directory in the Oracle Directory Integration Platform" for further instructions about initial bootstrapping

34

Synchronization with iPlanet Directory Server

This chapter explains how you can synchronize between Oracle Internet Directory and an iPlanet Directory Server by using the iPlanet Connector in the Directory Integration Platform.

This chapter contains these topics:

About the iPlanet Connector for Synchronizing between the Oracle Internet Directory Server and iPlanet Directory Server

The iPlanet Connector in the Directory Integration Platform enables you to:

- Import data into Oracle Internet Directory from an iPlanet Directory Server
- Export data from Oracle Internet Directory into an iPlanet Directory Server

You must configure a separate profile for each operation.

Synchronization is supported for iPlanet Directory Server release 4.13 and 5.0.

Configuring the Oracle Internet Directory Integration Solution for the iPlanet Directory Server

This section explains the tasks to configure the iPlanet Connector in the Directory Integration Platform. It contains these topics:

Task 1: Prepare Both Directories for Synchronization

1. Before synchronizing the two directories, ensure that the subscribed domains have equivalent user data in both directories. If the data is not equivalent, then migrate the most recent data to the other directory.

   See Also:

   - Appendix F, "Migrating Data from Other LDAP-Compliant Directories" for instructions on migrating data
   - iPlanet Directory Server documentation for instructions on migrating data to iPlanet Directory Server

2. At the end of migration, be sure that the change logging option for the Oracle directory server is set to the default, namely, TRUE. If it is set to FALSE, then

shut down the Oracle Internet Directory server and start with the changelog-enabled by using the **OID Control Utility**.

> **See Also:** a"Starting and Stopping an Oracle Directory Server Instance" on page A-44 for a description of the OID Control Utility

Similarly, verify that change logging is enabled in iPlanet Directory Server.

3. If the changelog is already enabled, note down the value of the `lastChangeNumber` attribute in Oracle Internet Directory and in the iPlanet Directory Server by using the following command for each directory:

```
ldapsearch -D SuperUserDn -w SuperUserPass -b "" -s base "objectclass=*"
lastchangenumber
```

In the next task, you use the value of the `lastChangeNumber` attribute in both directories to configure the following attributes in the integration profile:

- `orclLastAppliedChangeNumber`—to export from Oracle Internet Directory to iPlanet Directory Server
- `orclodipConDirLastAppliedChgNum`—to import from iPlanet Directory Server to Oracle Internet Directory

Task 2: Configure the Integration Profile for the Oracle Internet Directory Integration Solution for the iPlanet Directory Server

Integration profile templates for synchronizing with the iPlanet Directory Server are created in the Oracle Internet Directory Server as a part of the installation process. Deployment-specific parameters in the profile must be set before enabling synchronization.

Do this by using Oracle Directory Manager.

> **See Also:**
>
> - "Registration of Connectors into Oracle Directory Integration Platform" on page 29-5 for the required steps and a general description of each attribute you must set
> - Table 34–1 for a list and descriptions of the attribute information specific to the iPlanet Directory Server integration profile

Table 34–1 Attributes in the iPlanet Directory Server Integration Profile (Import/Export)

Attribute	Description
General Information	
Profile Name (`orclodipAgentName`)	The default value for the import profile is iPlanetImport. The default value for the export profile is iPlanetExport. This attribute is mandatory.
Profile Status (`orclodipAgentControl`)	You must set this value to ENABLE.
Profile Password (`orclodipProfilePassword`)	The default value is `welcome`. **Note:** For security reasons, change this password.
Synchronization Mode (`orclodipSynchronizationMode`)	Direction of synchronization between Oracle Internet Directory and the iPlanet Connector in the Directory Integration Platform. ■ IMPORT indicates importing changes from iPlanet Directory Server to Oracle Internet Directory. ■ EXPORT indicates exporting changes from Oracle Internet Directory to iPlanet Directory Server. This is already configured in the respective integration profiles.
Scheduling Interval (`orclodipSchedulingInterval`)	The default is 600 seconds. You can modify this to a different scheduling interval as per your requirement.
Maximum Number of Retries (`orclodipSyncRetryCount`)	Maximum number of times Oracle directory integration server tries to run the iPlanet Connector in the Directory Integration Platform in the event of a failure. The default is 5.
Execution Information	
Execution Command (`orclodipAgentExeCommad`)	This field must be empty.

Table 34–1 Attributes in the iPlanet Directory Server Integration Profile (Import/Export)

Attribute	Description
Connected Directory Account (`orclodipConDirAccessAccount`)	Valid user account on iPlanet Directory Server that the iPlanet Connector in the Directory Integration Platform uses to access iPlanet Directory Server. ■ If the changes are to be imported from iPlanet Directory Server to Oracle Internet Directory, then this user account should have read privilege in the iPlanet change log container. ■ If the changes in Oracle Internet Directory are to be exported to iPlanet Directory Server, then the user must have add/modify privileges to the synchronization domain. ■ **Note:** Create a user account in iPlanet exclusively for the iPlanet connector for synchronizing.
Connected Directory Account Password (`orclodipConDirAccessPassword`)	Password for the user account specified earlier for accessing iPlanet Directory Server.
Additional Config Info (`orclodipAgentConfigInfo`)	For the iPlanet Connector in the Directory Integration Platform, this attribute stores the iPlanet connector details to use its LDAP interface to synchronize with the iPlanet Directory Server. This information is already loaded in the integration profiles. Upload the file by using the `ldapUploadAgentFile.sh` tool. Do this for both import and export agents.
Interface Type (`orclodipInterfaceType`)	This attribute is set to LDAP.
Mapping Information	
Attribute Mapping Rules (`orclodipAttributeMappingRules`)	Store the mapping rules in a file by using the `ldapUpLoadAgentFile.sh` tool. **See Also:** "Task 3: Configure Mapping Rules" on page 34-7 for a detailed description of the entries in the mapping file

Table 34–1 Attributes in the iPlanet Directory Server Integration Profile (Import/Export)

Attribute	Description
Connected Directory Matching Filter (`orclodipConDirMatchingFilter`)	This attribute specifies the filter to apply to the iPlanet Directory Changelog. It is used in the import profile. The filter must be set in the import profile when both the import (iPlanetImport) and export (iPlanetExport) integration profiles are enabled, as follows: `Modifiersname != <connected directory account>` This prevents the same change from being exchanged between the two directories indefinitely.
OID Matching Filter	This attribute specifies the filter to apply to the Oracle Internet Directory Changelog container. It is used in the export profile. It must be set in the export profile when both the import (iPlanetImport) and export (iPlanetExport) integration profiles are enabled, as follows: `Modifiersname != orclodipagentname=iPlanetImport, cn=subscriber profile,cn= changelog subscriber,cn=oracle internet directory` This prevents the same change from being exchanged between the two directories indefinitely.
Status Information	
Synchronization Status (`orclodipSynchronizationStatus`)	Initially, this attribute has the value `Yet to be executed`. It is a read-only attribute.
Synchronization Errors (`orclodipSynchronizationErrors`)	Error messages, shown if the previous execution of the synchronization failed. This parameter is updated by Oracle directory integration server. It is a read-only attribute.
Connected Directory Last Applied Change Number (`orclodipConDirLastAppliedChgNum`)	The default value is `0`. Set this to the `lastchangenumber` value described in "Task 1: Prepare Both Directories for Synchronization" on page 34-2.

Table 34–1 Attributes in the iPlanet Directory Server Integration Profile (Import/Export)

Attribute	Description
OID Last Applied Change Number (orclLastAppliedChangeNumber)	The default value is `0`. Set this to the `lastchangenumber` value described in "Task 1: Prepare Both Directories for Synchronization" on page 34-2.
Last Execution Time (`orclodipLastExecutionTime`)	This attribute must be set to the `next execution time - the scheduling interval`
Last Successful Execution Time (`orclodipLastSuccessfulExecution Time`)	This attribute is a status attribute set to the last time the integration profile was executed successfully by the Directory Integration Server.

Task 3: Configure Mapping Rules

You can customize the attributes of the entries to be synchronized between iPlanet Directory Server and Oracle Internet Directory. You can also determine how to store the attribute values in the directories by using mapping rules.

A sample mapping file is provided in $*ORACLE_HOME*`/ldap/odi/conf/iPlanet.map.master`

This file must be loaded with the ldapUploadAgentFile.sh tool.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

See Also: "Mapping Rules and Formats" on page 29-9 for more details

Task 4: Configure Access Control

Set up appropriate ACLs allowing read, add, or modify access rights on the subscribed domains.

During import operations:

1. You would privilege the user `orclodipagentname=iPlanetImport,cn=subscriber profile,cn=changelog subscriber,cn=oracle internet directory` in Oracle Internet Directory to update the subscribed domain in Oracle Internet Directory.
2. The user specified by the Connected Directory Account attribute in the integration profile must have read access to the changelog container in the iPlanet Directory Server.

   For example, assuming that no ACLs are applied to the domain of interest, that is, the `Synchronization domain in OID`, the following LDIF sample can be used.

```
ACL in OID:
    dn: <Synchronization domain in OID>
    changetype: modify
    replace: orclaci
    orclaci: access to entry by
    "orclodipagentname=iPlanetImport,cn=subscriber profile,cn=changelog
    subscriber,cn=oracle internet directory" (browse,add,delete)
    orclaci: access to attr=(*) by
    "orclodipagentname=iPlanetImport,cn=subscriber profile,cn=changelog
    subscriber,cn=oracle internet directory" (read,search,write,compare)"
```

During export operations, the user specified by the Connected Directory Account attribute in the integration profile must have read access to the changelog contained in the iPlanet Directory Server.

> **See Also:** iPlanet Server documentation to apply ACLs on the iPlanet changelog container and the iPlanet subscribed domain

Task 5: Configure the Password Protection

To enable synchronization of any protected password attributes—for example, `userPassword`—configure the password hashing algorithm to be the same on both directories.

To set the hashing algorithm for the password in Oracle Internet Directory, use this command:

```
ldapmodify  -D SuperUserDn  -w SuperUserPass  << EOF
dn:
changetype: modify
```

```
replace: orclcryptoscheme
orclcryptoscheme: your_hashing_algorithm
```

See Also:

- "Protection of User Passwords for Directory Authentication" on page 11-7 for a list of the hashing algorithms that Oracle Internet Directory supports for password protection
- iPlanet Directory Server documentation for instructions on how to set the appropriate hashing algorithm for passwords in iPlanet Directory Server

Synchronizing Between Oracle Internet Directory and iPlanet Directory Server

This section contains these topics:

- Preparing for Synchronization
- The Synchronization Process

Preparing for Synchronization

To prepare for successful synchronization between Oracle Internet Directory and iPlanet Directory Server, verify the following:

- The Oracle directory integration server and iPlanet Directory Server are installed and running
- The configuration details are correct, as described in "Configuring the Oracle Internet Directory Integration Solution for the iPlanet Directory Server" on page 34-2
- The Oracle directory integration server on this host is registered to Oracle Internet Directory and running

The Synchronization Process

The synchronization process performs the following:

1. In an import operation, the iPlanet Connector in the Directory Integration Platform extracts all the changes from the source directory, namely, iPlanet Directory Server, based on the value specified in the `orclodipConDirLastAppliedChgNum` attribute, and applies them to Oracle

Internet Directory. Similarly, in an export operation, the iPlanet Connector in the Directory Integration Platform extracts all the changes from Oracle Internet Directory, based on the `orclodipLastChangeNumber`, and applies it to iPlanet Directory Server.

2. Once all the changes are read and applied, the appropriate attribute—either `orclodipConDirLastAppliedChgNum` or `orclodipLastAppliedChangeNumber`—is updated.
3. After the execution completion, Oracle directory integration server updates the execution status attributes.

Troubleshooting

The Oracle directory integration server stores error messages in the appropriate file, as described in Table on page 30-13.

Limitations in This Release

Oracle Internet Directory Release 9.0.2 does not support the synchronization of the schema and ACLs. If you are changing ACLs or the schema, then you must apply the changes manually.

A tool for schema synchronization, namely, SchemaSync, is available in Oracle Internet Directory Release 9.0.2.

> **See Also:** "SchemaSync Syntax" on page A-57 for information about the SchemaSync tool

35

Synchronization with Third-Party Metadirectory Solutions

Oracle Internet Directory uses change logs to enable synchronization with supported third party metadirectory solutions. This chapter describes how change log information is generated and how supporting solutions use that information. It tells you how to enable the directory integration agents of third-party metadirectory solutions so that they can synchronize with Oracle Internet Directory.

This chapter contains these topics:

About Change Logs

Oracle Internet Directory records each change as an entry in the change log container. A directory integration agent for the third-party directory retrieves changes from the change log container and applies them to the third-party directory. To retrieve these changes, the agent must subscribe to the Oracle Internet Directory change logs.

Each entry in the change log store has a change number. The agent keeps track of the number of the last change it applied, and it retrieves from Oracle Internet Directory only those changes with numbers greater than the last change it applied. For example, if the last change an agent retrieved had a number of 250, then subsequent changes it retrieves would have numbers greater than 250.

From the standpoint of the Oracle Directory Integration Platform, the agent for the third-party metadirectory solution is an **external agent**—that is, Oracle directory integration server does not provide mapping or scheduling services for it.

Note: If an agent is not subscribed to the Oracle Internet Directory change logs, and the first change it retrieves is more than one number higher than the last change it last applied, then some of the changes in the Oracle Internet Directory change log have been purged. In this case, the agent must read the entire Oracle Internet Directory to synchronize its copy with that in Oracle Internet Directory.

See Also: "About Connectors and Directory Integration Profiles" on page 29-2 for a conceptual discussion of directory integration agents, including external agents

Enabling External Agents to Synchronize with Oracle Internet Directory

To enable external agents to retrieve changes from Oracle Internet Directory, perform the tasks described in this section.

- Task 1: Perform Initial Bootstrapping
- Task 2: Create a Change Subscription Object in Oracle Internet Directory for the External Agent
- Task 3: Grant External Agents Access to the Oracle Internet Directory Change Log Object Container

Task 1: Perform Initial Bootstrapping

To bootstrap a directory to synchronize data between a local directory and Oracle Internet Directory, do the following:

1. Find the number of the last change recorded in Oracle Internet Directory. This number is contained in the DSE root attribute, `lastChangeNumber`.

   To find the number of the last change recorded in Oracle Internet Directory, use ldapsearch. Enter the following command:

   ```
   ldapsearch -h host_name -p port_number -s base -b "" 'objectclass=*'
   lastchangenumber
   ```

   If the change log does not contain change entries because they have been purged, then the last change number retrieved is `0` (zero).

2. Use ldifwrite to export data from Oracle Internet Directory into an LDIF file.

3. Convert the LDIF file to a format suitable to the client directory, then load it into the client directory.

Note: Initial bootstrapping is not required with a new installation of Oracle Internet Directory. In this case, the current change number of the newly installed Oracle Internet Directory is `0` (zero).

See Also: "ldifwrite Syntax" on page A-39 for instructions on using ldifwrite

Task 2: Create a Change Subscription Object in Oracle Internet Directory for the External Agent

To enable an external agent to synchronize with Oracle Internet Directory, you must create a change subscription object for it in Oracle Internet Directory. This gives the agent access to change log objects stored in Oracle Internet Directory.

About the Change Subscription Object

The change subscription object is an entry located under the following container in Oracle Internet Directory:

```
cn=Subscriber Profile,cn=ChangeLog Subscriber,cn=Oracle Internet Directory
```

This change subscription object provides a unique credential for an external agent to bind with Oracle Internet Directory and to retrieve changes from it. You associate the change subscription object with the auxiliary object class `orclChangeSubscriber`. This object class has several attributes, of which the following are mandatory:

- `userPassword`

  Password to be used by the directory when accessing the change log object in Oracle Internet Directory

- `orclLastAppliedChangeNumber`

  Number of the change applied during the last synchronization. This attribute allows the directory to retrieve only the changes in Oracle Internet Directory it has not already applied.

- `orclSubscriberDisable`

  Flag indicating whether the subscription of the external agent is enabled or disabled. A value of `1` indicates that it is disabled, and a value of `0` indicates that it is enabled.

Creating a Change Subscription Object

To create a change subscription object, use ldapadd. The following example uses an input file, named `add.ldif`, to create and enable a change subscription object, named `my_change_subscription_object`, under the container `cn=Subscriber Profile,cn=ChangeLog Subscriber,cn=`Oracle Internet Directory. The `orclLastAppliedChangeNumber` is the current change number in the directory before initial bootstrapping—in this example, 250.

- Edit file `add.ldif`:

```
dn: cn=my_change_subscription_object,cn=Subscriber Profile,cn=ChangeLog
Subscriber,cn=Oracle Internet Directory
userpassword: my_password
orclLastAppliedChangeNumber: 250
orclSubscriberDisable: 0
objectclass: orclChangeSubscriber
objectclass: top
```

- Add the entry:

```
ldapadd -h my_host -p 389 -f add.ldif
```

See Also: "Disabling and Deleting Change Subscription Objects" on page 35-7 for instructions on temporarily disabling change subscription objects or deleting them altogether

Task 3: Grant External Agents Access to the Oracle Internet Directory Change Log Object Container

Once you have created a change subscription object, you must grant it read access to the `cn=changeLog` entry in Oracle Internet Directory. You do this by adding it to the following group entry: `cn=odipgroup,cn=odi,cn=`Oracle Internet Directory.

The following example uses a file, named `add_to_group.ldif`, to add the entry created in the previous examples, namely, `my_change_subscription_object`, to this group entry `cn=odipgroup,cn=odi,cn=`Oracle Internet Directory`.`

- Edit the file `add_to_group.ldif`

```
dn: cn=odipgroup,cn=odi,cn=Oracle Internet Directory
changetype: modify
add: uniqueMember
uniqueMember: cn=my_change_subscription_object,cn=Subscriber Profile,
                cn=ChangeLog Subscriber,cn=Oracle Internet Directory
```

- Modify the entry:

```
ldapmodify -h my_ldap_host -p 389 -v -f add_to_group.ldif
```

The Synchronization Process

This section contains these topics:

- How a Connected Directory Retrieves Changes the First Time from Oracle Internet Directory
- How a Connected Directory Updates the orclLastAppliedChangeNumber Attribute in Oracle Internet Directory

How a Connected Directory Retrieves Changes the First Time from Oracle Internet Directory

In this example, a connected directory with a change subscription object named `my_change_subscription_object` acquires changes from Oracle Internet Directory.

```
ldapsearch -h my_host -p 389 -b "cn=changeLog" -s one
(&(objectclass=changeLogEntry)
(changeNumber >= orclLastAppliedChangeNumber )
( ! (modifiersname =cn=my_change_subscription_object,cn=Subscriber Profile,
     cn=ChangeLog Subscriber,cn=Oracle Internet Directory ) ) )
```

When the directory is retrieving changes for the first time, the value for `orclLastAppliedChangeNumber` is the number you set in "Task 2: Create a Change Subscription Object in Oracle Internet Directory for the External Agent" on page 35-3.

The argument `(!(modifiersname=client_bind_dn))` in the filter ensures that Oracle Internet Directory does not return changes made by the connected directory itself.

How a Connected Directory Updates the orclLastAppliedChangeNumber Attribute in Oracle Internet Directory

After retrieving changes from Oracle Internet Directory, the connected directory updates the `orclLastAppliedChangeNumber` attribute in its change subscription object in Oracle Internet Directory. This allows Oracle Internet Directory to purge changes that connected directories have already applied. It also enables the connected directory to retrieve only the most recent changes, ignoring those it has already applied.

This example uses an input file, `mod.ldif`, in which the connected directory has a change subscription object named `my_change_subscription_object`, and the last applied change number is 121. The connected directory updates `orclLastAppliedChangeNumber` in its change subscription object in Oracle Internet Directory as follows:

1. Edit `mod.ldif`:

```
dn: cn=my_change_subscription_object,cn=Subscriber Profile,
    cn=ChangeLog Subscriber,cn=Oracle Internet Directory
changetype:modify
replace: orclLastAppliedChangeNumber
orclLastAppliedChangeNumber: 121
```

2. Use ldapmodify to load the edited `mod.ldif` file:

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

   See Also: "Change Log Purging" on page 22-6 for information about purging changes according to change numbers

Disabling and Deleting Change Subscription Objects

You can temporarily disable an existing change subscription object, or delete it altogether. This section contains these topics:

- Disabling a Change Subscription Object
- Deleting a Change Subscription Object

Disabling a Change Subscription Object

If a change subscription object already exists for an agent, but you want to disable it temporarily, then set the `orcl`SubscriberDisable attribute to `1`. The following example uses an input file, `mod.ldif`, to disable a change subscription object.

- Edit file `mod.ldif`:

  ```
  dn: cn=my_change_subscription_object,cn=Subscriber Profile,
      cn=ChangeLog Subscriber,cn=Oracle Internet Directory
  changetype: modify
  replace: orclSubscriberDisable
  orclSubscriberDisable: 1
  ```

- Modify the entry:

  ```
  ldapmodify -h my_ldap_host -p 389 -v -f mod.ldif
  ```

Deleting a Change Subscription Object

To delete a change subscription object, use ldapdelete. Enter the following command:

```
ldapdelete -h ldap_host -p ldap_port
           "cn=my_change_subscription_object,cn=Subscriber Profile,
            cn=ChangeLog Subscriber,cn=Oracle Internet Directory"
```

36

The Oracle Directory Provisioning Integration Service

The Oracle Directory Provisioning Integration Service enables applications to receive provisioning information from Oracle Internet Directory.

This chapter contains these topics:

- About the Oracle Directory Provisioning Integration Service
- Managing the Oracle Directory Provisioning Integration Service Environment
- Security and the Oracle Directory Provisioning Integration Service
- Troubleshooting the Oracle Directory Provisioning Integration Service

> **See Also:** The chapter on developing provisioning-integrated applications in *Oracle Internet Directory Application Developer's Guide* in the Oracle9*i* Application Server documentation library

About the Oracle Directory Provisioning Integration Service

This section describes how the components of an Oracle Directory Provisioning Integration Service environment interact throughout the provisioning process. It contains these topics:

- About Provisioning
- How the Oracle Directory Provisioning Integration Service Retrieves Changes from Oracle Internet Directory
- How an Application Obtains Provisioning Information by Using the Oracle Directory Provisioning Integration Service

About Provisioning

Provisioning is the process of notifying an application whenever user or group data changes in Oracle Internet Directory. Provisioning events arise whenever any change occurs to a relevant user's or group's status or information. An application subscribes to provisioning when it is first installed by creating a provisioning profile in the directory. Subscription occurs once for each application.

Provisioning involves—but is not the same as—synchronization. At times, you may want to synchronize all entities in an application-specific directory with those in the central directory, but provision the application to receive notification only about some of them. For example, the directory for Oracle Human Resources typically contains data for all employees in an enterprise, and you would probably want to synchronize all of that data with the central directory. However, you might want to provision your application to receive notification only when members join or leave a particular group.

Provisioning Procedures

In a directory-enabled environment, provisioning involves:

1. Creating the user in the central directory
2. Enrolling the user in the application—that is, creating application-specific user accounts and entitlements
3. Synchronizing those accounts and entitlements with the central directory

For example, provisioning a user to access an e-mail application involves:

1. Creating the user in the central directory

2. Enrolling the user in the e-mail application. This involves setting up an e-mail account and quota for that user and creating the necessary public folders.
3. Synchronizing the user information in the e-mail application with that in the central directory

You can change user and group information from any of the following:

- Oracle Human Resources or other applications integrated with the Oracle Directory Integration Platform
- Oracle Directory Manager
- Oracle Enterprise Manager tools—for example, Enterprise Security Manager

User Enrollment in Applications

User enrollment in an application can happen either automatically or manually.

Automatic Enrollment An example of this is sometimes called "on-demand enrollment." Instead of continuously synchronizing with the central directory, the application creates the user footprint when the user first accesses the application. Oracle9*i*AS Single Sign-On enrolls a user accessing an application in this way.

Manual Enrollment The administrator provides application-specific information by using an application-specific administrative tool.

For example, you might want users to obtain their manager's approval before enrollment. In this case, rather than use on-demand enrollment, you might want the application administrator to enroll the user manually after the necessary approvals are complete.

Provisioning Information

Provisioning a user typically involves creating two kinds of information:

- Shared user metadata in Oracle Internet Directory

  This data includes the user's identity, credentials, profiles, and preferences. It is represented by standard directory user attributes—for example, mailing address or language preferences.

- Application-specific user data in the application

  This could include, for example, data in the user's e-mail message folder, or, for the calendaring application, the user's appointment data. It is typically

represented by using application-specific conventions either in the directory or in application-specific repositories.

How the Oracle Directory Provisioning Integration Service Retrieves Changes from Oracle Internet Directory

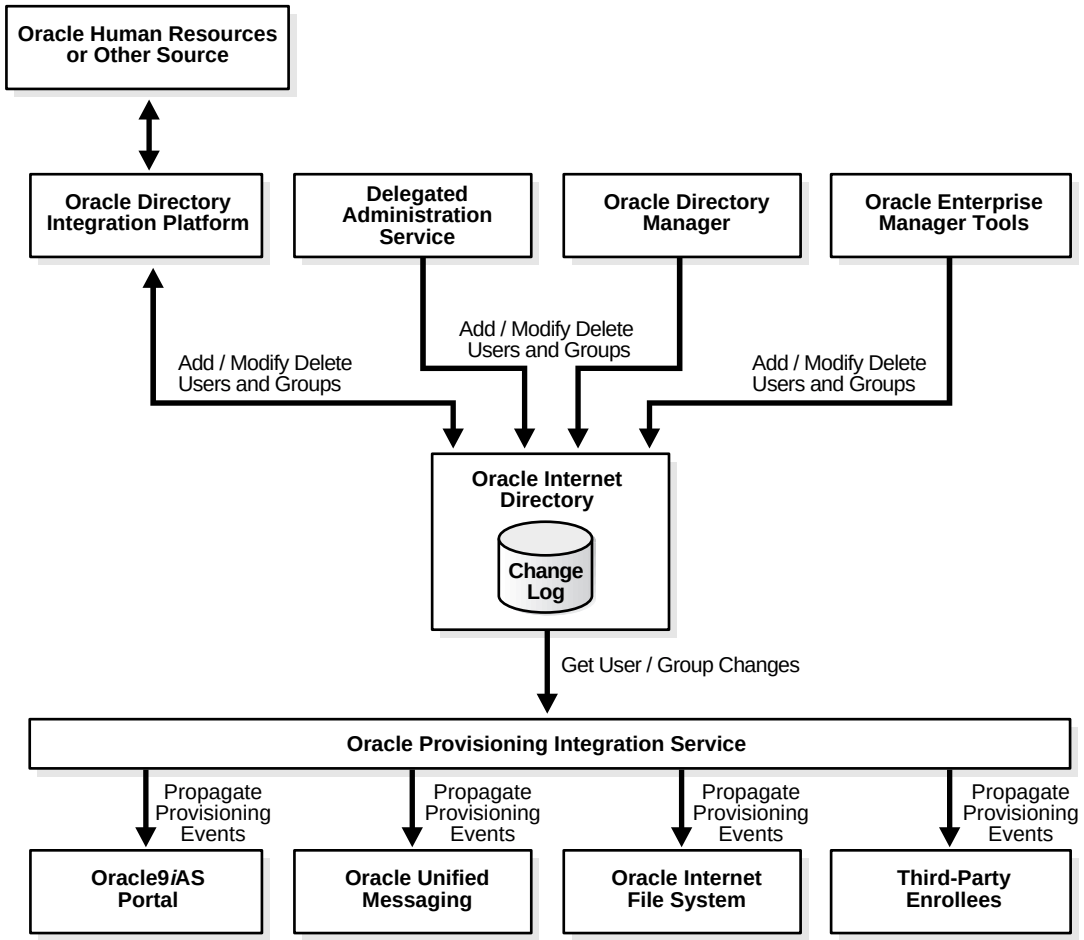
In an Oracle Directory Provisioning Integration Service environment:

- Oracle Internet Directory acts as the central repository for all user and group information
- Applications subscribe to receive the provisioning events by creating provisioning profiles in the directory
- The Oracle Directory Provisioning Integration Service monitors Oracle Internet Directory for any changes to user or group information, and conveys these changes to applications in the form of provisioning events

To retrieve changes from Oracle Internet Directory, the Oracle Directory Provisioning Integration Service subscribes to the Oracle Internet Directory change log. The changes in the change log are filtered so that only the needed changes get passed to the applications. For example, if an application is interested only in the events of a particular subtree, then the Oracle Directory Provisioning Integration Service notifies it of those changes only.

Figure 36–1 shows the relation between components in an Oracle Directory Provisioning Integration Service environment.

Figure 36–1 Typical Deployment of The Oracle Directory Provisioning Integration Service Environment



As Figure 36–1 shows:

- Oracle Internet Directory acts as the central repository for all user and group information
- Various components can add, modify, or delete user and group entries in Oracle Internet Directory. These components are:
  - Oracle Directory Integration Platform synchronizing with, for example, Oracle Human Resources or other repositories
  - The Delegated Administration Service
  - Oracle Directory Manager
  - Oracle Enterprise Manager tools—for example, the Enterprise Security Manager

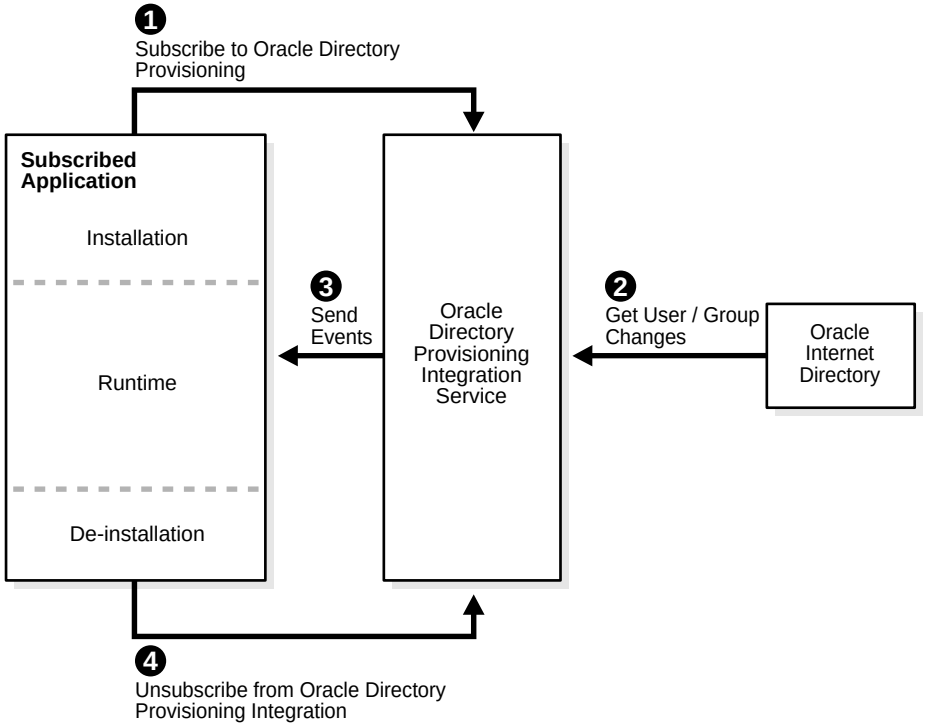
  The Oracle Internet Directory change log records these changes.
- The Oracle Directory Provisioning Integration Service retrieves changes to user and group information from Oracle Internet Directory and sends them to subscribed applications. In this example, the applications are Oracle9*i*AS Portal, Oracle Unified Messaging, Oracle Internet File System, and third-party enrollees.

How an Application Obtains Provisioning Information by Using the Oracle Directory Provisioning Integration Service

The Oracle Directory Provisioning Integration Service monitors Oracle Internet Directory for any changes to user or group information. It conveys these changes to applications in the form of provisioning events.

Figure 36–2 shows the life cycle of an application that obtains the provisioning events.

Figure 36–2 How an Application Obtains Provisioning Information by Using the Oracle Directory Provisioning Integration Service



1. Subscription to the Oracle Directory Provisioning Integration Service occurs in one of two ways:

   - The application subscribes itself automatically during application installation by using the Provisioning Subscription Tool
   - The administrator manually subscribes it by using the Provisioning Subscription Tool.

   The Provisioning Subscription Tool, `oidprovtool`, is invoked from any *`ORACLE_HOME`*`/bin`.The general pattern of invoking this tool is:

   ```
   oidprovtool param1=p1_value param2=p2_value param3=p3_value ...
   ```

See Also: "Provisioning Subscription Tool Syntax" on page A-30 for the Provisioning Subscription Tool parameters and the values they can take on

2. This tool, in turn, requests information that the application needs to subscribe to the Oracle Directory Provisioning Integration Service, including:
   - The host name and port number of the Oracle directory server instance
   - The user name and password of the Oracle Internet Directory user
   - Information to register the application with Oracle Internet Directory
   - Information to register the database connect information with Oracle Internet Directory
   - Information for the Oracle Directory Provisioning Integration Service to service the application—for example, the kind of changes required, or scheduling properties

   Once the necessary configuration information is in Oracle Internet Directory, the Oracle Directory Provisioning Integration Service periodically sends the changes to the application. The changes it sends are based on application-specific database connect information.
3. De-installation from Oracle Directory Provisioning Integration Service occurs in one of two ways:
   - The application de-installs itself automatically
   - The administrator manually unsubscribes it by using the Provisioning Subscription Tool

Managing the Oracle Directory Provisioning Integration Service Environment

This section contains these topics:

Overview: Deploying the Oracle Directory Provisioning Integration Service

To deploy the Oracle Directory Provisioning Integration Service, you perform these general steps:

1. Install Oracle Internet Directory—which includes the Oracle Directory Integration Platform—and load user information into it.
2. Install the applications and, when the Provisioning Subscription Tool prompts, supply the information that the applications need to subscribe to the Oracle Directory Provisioning Integration Service. This enables them to receive provisioning events.
3. Periodically monitor the status of the provisioning event propagation for each application.

Managing the Oracle Directory Provisioning Integration Service

This section describes:

- How to manage the Oracle directory integration server
- How to manage provisioning profiles

Managing the Oracle Directory Integration Server

The Oracle directory integration server runs the Oracle Directory Provisioning Integration Service to propagate provisioning events to subscribed applications.

Note: When the Oracle directory integration server is invoked in the default mode, it supports only the Oracle Directory Provisioning Integration Service, and not the Oracle Directory Synchronization Service.

> **See Also:** "Managing the Oracle Directory Integration Server" on page 30-7 for instructions about managing the Oracle directory integration server

Managing Provisioning Profiles

Use the Provisioning Subscription Tool to perform these activities:

- Create a new provisioning profile. A new provisioning profile is created and set to the enabled state so that the Oracle Directory Integration Platform can process it
- Disable an existing provisioning profile
- Enabled a disabled provisioning profile
- Delete an existing provisioning profile
- Get the current status of a given provisioning profile
- Clear all of the errors in an existing provisioning profile

Use the OID Server Manageability functionality in the Oracle Enterprise Manager to monitor provisioning profiles.

> **See Also:** the following for more details:
>
> - "Provisioning Subscription Tool Syntax" on page A-30
> - *Oracle Enterprise Manager Concepts Guide*
> - *Oracle Enterprise Manager Administrator's Guide*
> - Oracle Enterprise Manager online help

Security and the Oracle Directory Provisioning Integration Service

This section describes the principal entities involved in the provisioning integration process and the directory privileges that they need to complete various operations. It contains these topics:

- The Need to Control Access to Provisioning Profiles
- Entities Needing Access
- Entry-Level Privileges Granted to Entities
- Attribute Level Privileges Granted to Entities

The Need to Control Access to Provisioning Profiles

There are important reasons to control access to the provisioning profiles of applications:

- These profiles contain confidential information about the application—information that should not be viewable by unauthorized directory entities
- Providing provisioning events to applications consumes system resources. The number of those who can provision applications should be limited.

Entities Needing Access

The access that you grant to entities to operate on profiles depends on the delegation needs of the applications. Entities that need controlled access to the provisioning profiles are:

- The Oracle Directory Integration Server group—that is, `cn=odisgroup,cn=odi,cn=oracle internet directory`
- Provisioning administrators—that is, `cn=Provisioning Admins, cn=Provisioning Profiles...`
- Application Entities—that is, users for whom the value of the `orclGUID` attribute is `orclODIPProvisioningAppGUID`)
- Provisioning profiles—that is, users identified by the DN of the provisioning profiles
- All other users

Applications do not automatically have the rights to create provisioning profiles. Rather, only an LDAP identity with privileges to administer provisioning profiles can create them.

Provisioning administrators are modeled as a group and can perform any operation on the provisioning profiles. All other identities have lesser privileges.

Entry-Level Privileges Granted to Entities

Table 36–1 shows the entry-level privileges granted to each entity.

Table 36–1 Entry-Level Privileges

User Category	Browse	Add	Delete	Explanation
Oracle Directory Integration Server	Yes	No	Yes	Oracle Directory Integration Servers need to: ▪ Browse all provisioning profiles ▪ Delete some rogue provisioning profiles that the applications did not bother to delete However, Oracle Directory Integration Servers should not have access to add new provisioning profiles.
Provisioning administrators	Yes	Yes	Yes	The provisioning administrators group requires all privileges.
Application entities	Yes	No	Yes	Application entities themselves cannot create provisioning profiles, nor can they view another application's profiles. However, once a profile has been created, they can browse, modify, and delete their own profiles.
Provisioning profiles	Yes	No	No	Provisioning profiles also have an identity in the directory. For Release 9.0.2, this identity is not used, and hence it has the privilege only to perform a self-browse.
All other users	No	No	No	All other users should not be able to either browse, add, or delete provisioning profiles.

Attribute Level Privileges Granted to Entities

Provisioning profiles contain security-sensitive attributes that need protection from unauthorized access. Table 36–2 describes them.

Table 36–2 Attribute Level Privileges Granted to Entities

Attribute	Description
`userpassword`	Stores the directory user password
`orclPasswordAttribute`	Stores the clear text version of the directory user password
`orclODIPProfileInterfaceConnectInformation`	Stores details of the connection information to the target application, including the password to the target system
`orclODIPProfileInterfaceAdditionalInformation`	Stores any interface-specific information

Table 36–3 describes the access control for the secure attributes for the main entities operating on the provisioning profiles.

Table 36–3 Access Control for Secure Attributes

User Category	Read	Write	Search	Compare	Explanation
Oracle Directory Integration Servers	Yes	No	Yes	Yes	Oracle Directory Integration Servers need access to the secure attributes to complete their processing cycles. However, they do not need write access to them because these attributes should only be controlled by the Application Entities as well as Provisioning Admins.
Provisioning administrators	Yes	Yes	Yes	Yes	Provisioning administrators must be able to solve integration problems, and this requires full access to the secure attributes.

Table 36–3 Access Control for Secure Attributes

User Category	Read	Write	Search	Compare	Explanation
Application entities	Yes	Yes	Yes	Yes	Application entities are the real owners of the secure attributes, and this requires full access to the secure attributes.
Provisioning profiles	Yes	No	Yes	No	Provisioning profiles do not need to write or compare these attributes. As a result, they need only read and search privileges.
All other users	No	No	No	No	All other users receive no privileges.

The following table shows the access control for all other attributes in the provisioning profiles.

Table 36–4 Access Control for All other Attributes

User Category	Read	Write	Search	Compare
Oracle Directory Integration Servers	Yes	Yes	Yes	Yes
Provisioning administrators	Yes	Yes	Yes	Yes
Application entities	Yes	Yes	Yes	Yes
Provisioning profiles	Yes	Yes	Yes	Yes
All other users	No	No	No	No

Unlike secure attributes, the other attributes require a less strict access control. Full access is given to all entities involved in the provisioning process: Oracle Directory Integration Servers, provisioning administrators, application entities, and provisioning profiles. All other users receive no access to these attributes.

Troubleshooting the Oracle Directory Provisioning Integration Service

This section lists and describes the provisioning error messages you may see, and discusses actions to resolve them. These messages appear in the provisioning error messages attribute.

Table 36–5 Provisioning Error Messages

Message	Reason	Remedial Action
LDAP Connection Failure	The Oracle Directory Integration Platform failed to connect to the directory server.	Check the connection to the directory server. **See Also:** "Viewing Active Server Instance Information" on page 5-36 to get information about directory server connections
LDAP Authentication Failure	The provisioning profile is not able to connect to the LDAP Server as administrator	Verify Oracle directory integration server entry in the directory. Re-register the Oracle directory integration server by using `odisrvreg`. **See Also:** "Registering the Oracle Directory Integration Server" on page 30-2
Initialization Failure	Problem in connecting to the directory server using JNDI.	Look at the trace file for stack trace in `$ORACLE_HOME/ldap/odi/log/PROFILE_NAME.trc`
Database Connection Failure	Problem connecting to the database with the given account information. Either the database is not running or there is an authentication problem.	Look at the trace file for stack trace in `$ORACLE_HOME/ldap/odi/log/PROFILE_NAME.trc`
Exception while calling SQL Operation	Problem in executing the package.	Verify the package usability.

Part IX

Appendixes

This part contains these appendixes:

A

Syntax for LDIF and Command-Line Tools

This appendix provides syntax, usage notes, and examples for **LDAP Data Interchange Format (LDIF)** and LDAP command-line tools. It contains these topics:

LDAP Data Interchange Format (LDIF) Syntax

The standardized file format for directory entries is as follows:

```
dn: distinguished_name
attribute_type: attribute_value
.
.
.
objectClass: object_class_value
.
.
.
```

Property	Value	Description
dn:	*RDN,RDN,RDN,* ...	Separate RDNs with commas.
attribute:	*attribute_value*	This line repeats for every attribute in the entry, and for every attribute value in multi-valued attributes.
objectClass:	*object_class_ value*	This line repeats for every object class.

The following example shows a file entry for an employee. The first line contains the DN. The lines that follow the DN begin with the mnemonic for an attribute, followed by the value to be associated with that attribute. Note that each entry ends with lines defining the object classes for the entry.

```
dn: cn=Suzie Smith,ou=Server Technology,o=Acme, c=US
cn: Suzie Smith
cn: SuzieS
sn: Smith
email: ssmith@us.Acme.com
telephoneNumber: 69332
photo: /ORACLE_HOME/empdir/photog/ssmith.jpg
objectClass: organizationalPerson
objectClass: person
objectClass: top
```

The next example shows a file entry for an organization:

```
dn: o=Acme,c=US
o: Acme
ou: Financial Applications
objectClass: organization
objectClass: top
```

LDIF Formatting Notes

A list of formatting rules follows. This list is not exhaustive.

- All mandatory attributes belonging to an entry being added must be included with non-null values in the LDIF file.

  Tip: To see the mandatory and optional attribute types for an object class, use Oracle Directory Manager. See "Viewing Properties of Object Classes by Using Oracle Directory Manager" on page 6-9.

- Non-printing characters and tabs are represented in attribute values by base-64 encoding.
- The entries in your file must be separated from each other by a blank line.
- A file must contain at least one entry.
- Lines can be continued to the next line by beginning the continuation line with a space or a tab.
- Add a blank line between separate entries.
- Reference binary files, such as photographs, with the absolute address of the file, preceded by a forward slash ("/").
- The DN contains the full, unique directory address for the object.
- The lines listed after the DN contain both the attributes and their values. DNs and attributes used in the input file must match the existing structure of the DIT. Do not use attributes in the input file that you have not implemented in your DIT.
- Sequence the entries in an LDIF file so that the DIT is created from the top down. If an entry relies on an earlier entry for its DN, make sure that the earlier entry is added before its child entry.
- When you define schema within an LDIF file, insert a white space between the opening parenthesis and the beginning of the text, and between the end of the text and the ending parenthesis.

See Also:

- The various resources listed in "Related Documentation" on page xli for a complete list of LDIF formatting rules
- "Using Globalization Support with LDIF Files" on page 8-3

Command-Line Tools Syntax

This section tells you how to use the following tools:

- ldapadd Syntax
- ldapaddmt Syntax
- ldapbind Syntax
- ldapcompare Syntax
- ldapdelete Syntax
- ldapmoddn Syntax
- ldapmodify Syntax
- ldapmodifymt Syntax
- ldapsearch Syntax
- ldapUploadAgentFile.sh Syntax
- ldapCreateConn.sh Syntax
- StopOdiServer.sh Syntax

ldapadd Syntax

The ldapadd command-line tool enables you to add entries, their object classes, attributes, and values to the directory. To add attributes to an existing entry, use the ldapmodify command, explained in "ldapmodify Syntax" on page A-15.

See Also: "Adding Configuration Set Entries by Using ldapadd" on page 5-11 for an explanation of using ldapadd to configure a server with an input file

ldapadd uses this syntax:

```
ldapadd [arguments] -f filename
```

where *filename* is the name of an LDIF file written with the specifications explained in the section "LDAP Data Interchange Format (LDIF) Syntax" on page A-2.

The following example adds the entry specified in the LDIF file `my_ldif_file.ldi`:

```
ldapadd -p 389 -h myhost -f my_ldif_file.ldi
```

Optional Arguments	Description
-b	Specifies that you have included binary file names in the file, which are preceded by a forward slash character. The tool retrieves the actual values from the file referenced.
-c	Tells ldapadd to proceed in spite of errors. The errors will be reported. (If you do not use this option, ldapadd stops when it encounters an error.)
-D "*binddn"*	When authenticating to the directory, specifies doing so as the entry specified in *binddn*. Use this with the `-w` *password* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 8, "Globalization Support in the Directory".
-f *filename*	Specifies the input name of the LDIF format import data file. For a detailed explanation of how to format an LDIF file, see "LDAP Data Interchange Format (LDIF) Syntax" on page A-2.
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
-K	Same as `-k`, but performs only the first step of the Kerberos bind
-k	Authenticates using Kerberos authentication instead of simple authentication. To enable this option, you must compile with KERBEROS defined. You must already have a valid ticket granting ticket.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-n	Shows what would occur without actually performing the operation
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.

Optional Arguments	Description
-p *directory_server_port_number*	Connects to the directory on TCP port *directory_server_port_number*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-v	Specifies verbose mode
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Provides the password required to connect

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapaddmt Syntax

ldapaddmt is like ldapadd: It enables you to add entries, their object classes, attributes, and values to the directory. It is unlike ldapadd in that it supports multiple threads for adding entries concurrently.

While it is processing LDIF entries, ldapaddmt logs errors in the `add.log` file in the current directory.

ldapaddmt uses this syntax:

`ldapaddmt -T` *`number_of_threads`* `-h` *`host`* `-p` *`port`* `-f` *`filename`*

where *`filename`* is the name of an LDIF file written with the specifications explained in the section "LDAP Data Interchange Format (LDIF) Syntax" on page A-2.

The following example uses five concurrent threads to process the entries in the file `myentries.ldif`.

```
ldapaddmt -T 5 -h node1 -p 3000 -f myentries.ldif
```

> **Note:** Increasing the number of concurrent threads improves the rate at which LDIF entries are created, but consumes more system resources.

Optional Arguments	Description
-b	Specifies that you have included binary file names in the data file, which are preceded by a forward slash character. The tool retrieves the actual values from the file referenced.
-c	Tells the tool to proceed in spite of errors. The errors will be reported. (If you do not use this option, the tool stops when it encounters an error.)
-D "*binddn"*	When authenticating to the directory, specifies doing so as the entry is specified in *binddn*. Use this with the `-w` *`password`* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 8, "Globalization Support in the Directory"
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
-K	Same as -k, but performs only the first step of the kerberos bind
-k	Authenticates using Kerberos authentication instead of simple authentication. To enable this option, you must compile with KERBEROS defined. You must already have a valid ticket granting ticket.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-n	Shows what would occur without actually performing the operation.
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.

Optional Arguments	Description
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-T	Sets the number of threads for concurrently processing entries
-U *SSLAuth*	Specifies SSL Authentication Mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-v	Specifies verbose mode
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Provides the password required to connect

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapbind Syntax

The ldapbind command-line tool enables you to see whether you can authenticate a client to a server.

ldapbind uses this syntax:

`ldapbind [`*`arguments`*`]`

Optional Arguments	Description
-D "*binddn*"	When authenticating to the directory, specifies doing so as the entry specified in *binddn*. Use this with the `-w` *password* option.
-E "*.character_set*"	Specifies native character set encoding. See Chapter 8, "Globalization Support in the Directory".

Optional Arguments	Description
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
-n	Shows what would occur without actually performing the operation
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies the wallet password required for one-way or two-way SSL connections
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Provides the password required to connect

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapcompare Syntax

The ldapcompare command-line tool enables you to match attribute values you specify in the command line with the attribute values in the directory entry.

ldapcompare uses this syntax:

`ldapcompare [`*`arguments`*`]`

The following example tells you whether `Person Nine`'s title is `associate`.

```
ldapcompare -p 389 -h myhost -b "cn=Person Nine,ou=EuroSInet Suite,o=IMC,c=US"
-a title -v associate
```

Mandatory Arguments	Description
-a *attribute name*	Specifies the attribute on which to perform the compare
-b "*basedn*"	Specifies the distinguished name of the entry on which to perform the compare
-v *attribute value*	Specifies the attribute value to compare

Optional Arguments	Description
-D *binddn*	When authenticating to the directory, specifies doing so as the entry is specified in *binddn*. Use this with the `-w` *password* option.
-d *debug-level*	Sets the debugging level. See "Setting Debug Logging Levels by Using the OID Control Utility" on page 5-27.
-E "*character_set*"	Specifies native character set encoding. See Chapter 8, "Globalization Support in the Directory".
-f *filename*	Specifies the input filename
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Provides the password required to connect

Optional Arguments	Description

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapdelete Syntax

The ldapdelete command-line tool enables you to remove entire entries from the directory that you specify in the command line.

ldapdelete uses this syntax:

`ldapdelete [`*`arguments`*`] ["`*`entry_DN`*`" | `*`-f input_filename`*`]`

> **Note:** If you specify the entry DN, then do not use the `-f` option.

The following example uses port 389 on a host named myhost.

```
ldapdelete -p 389 -h myhost "ou=EuroSInet Suite, o=IMC, c=US"
```

Optional Argument	Description
-D "*binddn"*	When authenticating to the directory, uses a full DN for the *binddn* parameter; typically used with the `-w` *password* option.
-d *debug-level*	Sets the debugging level. See "Setting Debug Logging Levels by Using the OID Control Utility" on page 5-27.
-E "*character_set*"	Specifies native character set encoding. See Chapter 8, "Globalization Support in the Directory".
-f *input_filename*	Specifies the input filename
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
-k	Authenticates using authentication instead of simple authentication. To enable this option, you must compile with Kerberos defined. You must already have a valid ticket granting ticket.

Optional Argument	Description
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-n	Shows what would be done, but doesn't actually delete
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-v	Specifies verbose mode
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Provides the password required to connect.

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapmoddn Syntax

The ldapmoddn command-line tool enables you to modify the DN or RDN of an entry.

ldapmoddn uses this syntax:

```
ldapmoddn [arguments]
```

The following example uses ldapmoddn to modify the RDN component of a DN from `"cn=mary smith"` to `"cn=mary jones"`. It uses port 389, and a host named `myhost`.

```
ldapmoddn -p 389 -h myhost -b "cn=mary smith,dc=Americas,dc=imc,dc=com" -R
"cn=mary jones"
```

Mandatory Argument	Description
-b "*basedn*"	Specifies DN of the entry to be moved

Optional Argument	Description
-D "*binddn*"	When authenticating to the directory, do so as the entry is specified in *binddn*. Use this with the `-w` *password* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 8, "Globalization Support in the Directory".
-f *filename*	Specifies the input filename
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-N *newparent*	Specifies new parent of the RDN
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).

Optional Argument	Description
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-r	Specifies that the old RDN is not retained as a value in the modified entry. If this argument is not included, the old RDN is retained as an attribute in the modified entry.
-R *newrdn*	Specifies new RDN
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Provides the password required to connect.

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapmodify Syntax

The ldapmodify tool enables you to act on attributes.

ldapmodify uses this syntax:

`ldapmodify [arguments] -f filename`

where *filename* is the name of an LDIF file written with the specifications explained the section "LDAP Data Interchange Format (LDIF) Syntax" on page A-2.

The list of arguments in the following table is not exhaustive.

Optional Argument	Description
-a	Denotes that entries are to be added, and that the input file is in LDIF format.
-b	Specifies that you have included binary file names in the data file, which are preceded by a forward slash character.
-c	Tells ldapmodify to proceed in spite of errors. The errors will be reported. (If you do not use this option, ldapmodify stops when it encounters an error.)
-D "*binddn*"	When authenticating to the directory, specifies doing so as the entry is specified in *binddn*. Use this with the `-w` *password* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 8, "Globalization Support in the Directory".
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-n	Shows what would occur without actually performing the operation.
`-o log_file_name`	Can be used with the `-c` option to write the erroneous LDIF entries in the logfile. You must specify the absolute path for the log file name.
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).

Optional Argument	Description
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-v	Specifies verbose mode
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Overrides the default, unauthenticated, null bind. To force authentication, use this option with the `-D` option.

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

To run `modify`, `delete`, and `modifyrdn` operations using the `-f` flag, use LDIF for the input file format (see "LDAP Data Interchange Format (LDIF) Syntax" on page A-2) with the specifications noted below:

If you are making several modifications, then, between each modification you enter, add a line that contains a hyphen (-) only. For example:

```
dn: cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
changetype: modify
add: work-phone
work-phone: 510/506-7000
work-phone: 510/506-7001
-
delete: home-fax
```

Unnecessary space characters in the LDIF input file, such as a space at the end of an attribute value, will cause the LDAP operations to fail.

Line 1: Every change record has, as its first line, the literal `dn:` followed by the DN value for the entry, for example:

```
dn:cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
```

Line 2: Every change record has, as its second line, the literal `changetype:` followed by the type of change (`add, delete, modify, modrdn`), for example:

```
changetype: modify
```

or

```
changetype: modrdn
```

Format the remainder of each record according to the following requirements for each type of change:

- `changetype: add`

  Uses LDIF format (see "LDAP Data Interchange Format (LDIF) Syntax" on page A-2).

- `changetype: modify`

  The lines that follow this changetype consist of changes to attributes belonging to the entry that you identified in Line 1 above. You can specify three different types of attribute modifications—add, delete, and replace—which are explained next:

  – **Add attribute values**. This option to changetype modify adds more values to an existing multi-valued attribute. If the attribute does not exist, it adds the new attribute with the specified values:

    ```
    add: attribute name
    attribute name: value1
    attribute name: value2...
    ```

    For example:

    ```
    dn:cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
    changetype: modify
    add: work-phone
    work-phone: 510/506-7000
    work-phone: 510/506-7001
    ```

– **Delete values**. If you supply only the *delete* line, all the values for the specified attribute are deleted. Otherwise, if you specify an attribute line, you can delete specific values from the attribute:

```
delete: attribute name
[attribute name: value1]
```

For example:

```
dn: cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
changetype: modify
delete: home-fax
```

– **Replace values.** Use this option to replace all the values belonging to an attribute with the new, specified set:

```
replace: attribute name
[attribute name: value1 ...]
```

If you do not provide any attributes with `replace`, then the directory adds an empty set. It then interprets the empty set as a delete request, and complies by deleting the attribute from the entry. This is useful if you want to delete attributes that may or may not exist.

For example:

```
dn: cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
changetype: modify
replace: work-phone
work-phone: 510/506-7002
```

* `changetype:delete`

  This change type deletes entries. It requires no further input, since you identified the entry in Line 1 and specified a changetype of delete in Line 2.

  For example:

  ```
  dn: cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
  changetype: delete
  ```

* `changetype:modrdn`

  The line following the change type provides the new relative distinguished name using this format:

  ```
  newrdn: RDN
  ```

For example:

```
dn: cn=Barbara Fritchy,ou=Sales,o=Oracle,c=US
changetype: modrdn
newrdn: cn=Barbara Fritchy-Blomberg
```

Example: Using ldapmodify to Add an Attribute

This example adds a new attribute called `myAttr`. The LDIF file for this operation is:

```
dn: cn=subschemasubentry
changetype: modify
add: attributetypes
attributetypes: (1.2.3.4.5.6.7 NAME 'myAttr' DESC 'New attribute definition'
EQUALITY caseIgnoreMatch SYNTAX
'1.3.6.1.4.1.1466.115.121.1.15' )
```

On the first line, enter the DN specifying where this new attribute is to be located. All attributes and object classes they are stored in `cn=subschemasubentry`.

The second and third lines show the proper format for adding a new attribute.

The last line is the attribute definition itself. The first part of this is the object identifier number: `1.2.3.4.5.6.7`. It must be unique among all other object classes and attributes. Next is the `NAME` of the attribute. In this case the attribute NAME is `myAttr`. It must be surrounded by single quotes. Next is a description of the attribute. Enter whatever description you want between single quotes. At the end of this attribute definition in this example are optional formatting rules to the attribute. In this case we are adding a matching rule of `EQUALITY caseIgnoreMatch` and a SYNTAX of `Directory String`. This example uses the object ID number of 1.3.6.1.4.1.1466.115.121.1.15 instead of the SYNTAXES name which is "Directory String".

Put your attribute information in a file formatted like this example. Then run the following command to add the attribute to the schema of your Oracle directory server.

```
ldapmodify -h yourhostname -p 389 -D "orcladmin" -w "welcome" -v -f
/tmp/newattr.ldif
```

This ldapmodify command assumes that your Oracle directory server is running on port 389, that your super user account name is `orcladmin`, that your super user

password is `welcome` and that the name of your LDIF file is `newattr.ldif`. Substitute the host name of your computer where you see *`yourhostname`*.

If you are not in the directory where the LDIF file is located, then you must enter the full directory path to the file at the end of your command. This example assumes that your LDIF file is located in the `/tmp` directory.

ldapmodifymt Syntax

The ldapmodifymt command-line tool enables you to modify several entries concurrently.

ldapmodifymt uses this syntax:

```
ldapmodifymt -T number_of_threads [arguments] -f filename
```

where *`filename`* is the name of an LDIF file written with the specifications explained the section "LDAP Data Interchange Format (LDIF) Syntax" on page A-2.

> **See Also:** "ldapmodify Syntax" on page A-15 for additional formatting specifications used by ldapmodifymt

The following example uses five concurrent threads to modify the entries in the file `myentries.ldif`.

```
ldapmodifymt -T 5 -h node1 -p 3000 -f myentries.ldif
```

> **Note:** The ldapmodifymt tool logs error messages in the file `add.log`, which is located in the directory where you are running the command.

Optional Argument	Description
-a	Denotes that entries are to be added, and that the input file is in LDIF format. (If you are running ldapadd, this flag is not required.)
-b	Specifies that you have included binary file names in the data file, which are preceded by a forward slash character.
-c	Tells ldapmodify to proceed in spite of errors. The errors will be reported. (If you do not use this option, ldapmodify stops when it encounters an error.)

Optional Argument	Description
-D "*binddn"*	When authenticating to the directory, specifies doing so as the entry is specified in binddn. Use this with the `-w` *password* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter 8, "Globalization Support in the Directory".
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-n	Shows what would occur without actually performing the operation.
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-T	Sets the number of threads for concurrently processing entries
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-v	Specifies verbose mode
-V *ldap_version*	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
-w *password*	Overrides the default, unauthenticated, null bind. To force authentication, use this option with the -D option.

Optional Argument	Description

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

ldapsearch Syntax

The ldapsearch command-line tool enables you to search for and retrieve specific entries in the directory.

ldapsearch uses this syntax:

`ldapsearch [`*`arguments`*`] `*`filter`*` [`*`attributes`*`]`

The *filter* format must be compliant with RFC-2254.

> **See Also:** http://www.ietf.org/rfc/rfc2254.txt for further information about the standard for the filter format

Separate attributes with a space. If you do not list any attributes, all attributes are retrieved.

> **Note:** The ldapsearch tool does not generate LDIF output by default. To generate LDIF output from the ldapsearch command-line tool, use the `-L` flag.

Mandatory Argument	Description
-b "*basedn*"	Specifies the base DN for the search
-s *scope*	Specifies search scope: base, one, or sub

Optional Argument	Description
-A	Retrieves attribute names only (no values)

Optional Argument	Description
-a *deref*	Specifies alias dereferencing: never, always, search, or find
-B	Allows printing of non-ASCII values
-D "*binddn"*	When authenticating to the directory, specifies doing so as the entry specified in *binddn*. Use this with the `-w` *password* option.
-d *debug level*	Sets debugging level to the level specified (see Table 5–1 on page 5-28)
-E "*character_set*"	Specifies native character set encoding. See Chapter 8, "Globalization Support in the Directory".
-f *file*	Performs sequence of searches listed in *file*
-F sep	Prints ‘`sep`’ instead of ‘`=`’ between attribute names and values
-h *ldaphost*	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
-L	Prints entries in LDIF format (`-B` is implied)
-l *timelimit*	Specifies maximum time (in seconds) to wait for ldapsearch command to complete
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
-n	Shows what would be done without actually searching
-O *ref_hop_limit*	Specifies the number of referral hops that a client should process. The default value is 5.
-p *ldapport*	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
-P *wallet_password*	Specifies wallet password required for one-way or two-way SSL connections
-S *attr*	Sorts the results by attribute *attr*
-t	Writes to files in `/tmp`
-u	Includes user friendly entry names in the output
-U *SSLAuth*	Specifies the SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required

Optional Argument	Description
-v	Specifies verbose mode
-w *passwd*	Specifies bind passwd for simple authentication

-W *wallet_location*	Specifies wallet location required for one-way or two-way SSL connections. For example, on Solaris, you could set this parameter as follows:
	`-W "file:/home/my_dir/my_wallet"`
	On Windows NT, you could set this parameter as follows:
	`-W "file:C:\my_dir\my_wallet"`

-z *sizelimit*	Specifies maximum number of entries to retrieve

Examples of ldapsearch Filters

Study the following examples to see how to build your own search commands.

Example 1: Base Object Search The following example performs a base-level search on the directory from the root.

```
ldapsearch -p 389 -h myhost -b "" -s base -v "objectclass=*"
```

- `-b` specifies base DN for the search, root in this case.
- `-s` specifies whether the search is a base search (`base`), one level search (`one`) or subtree search (`sub`).
- `"objectclass=*"` specifies the filter for search.

Example 2: One-Level Search The following example performs a one level search starting at `"ou=HR, ou=Americas, o=IMC, c=US"`.

```
ldapsearch -p 389 -h myhost -b "ou=HR, ou=Americas, o=IMC, c=US" -s one -v
"objectclass=*"
```

Example 3: Subtree Search The following example performs a subtree search and returns all entries having a DN starting with `"cn=us"`.

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub -v "cn=Person*"
```

Example 4: Search Using Size Limit The following example actually retrieves only two entries, even if there are more than two matches.

```
ldapsearch -h myhost -p 389 -z 2 -b "ou=Benefits,ou=HR,ou=Americas,o=IMC,c=US"
```

```
-s one "objectclass=*"
```

Example 5: Search with Required Attributes The following example returns only the DN attribute values of the matching entries:

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub -v "objectclass=*" dn
```

The following example retrieves only the distinguished name along with the surname (sn) and description (description) attribute values:

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub -v "cn=Person*" dn sn description
```

Example 6: Search for Entries with Attribute Options The following example retrieves entries with common name (cn) attributes that have an option specifying a language code attribute option. This particular example retrieves entries in which the common names are in French and begin with the letter R.

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub "cn;lang-fr=R*"
```

Suppose that, in the entry for John, no value is set for the cn;lang-it language code attribute option. In this case, the following example does not return John's entry:

```
ldapsearch -p 389 -h myhost -b "c=us" -s sub "cn;lang-it=Giovanni"
```

Example 7: Searching for All User Attributes and Specified Operational Attributes The following example retrieves all user attributes and the createtimestamp and orclguid operational attributes:

```
ldapsearch -p 389 -h myhost -b "ou=Benefits,ou=HR,ou=Americas,o=IMC,c=US" -s sub
"cn=Person*" * createtimestamp orclguid
```

The following example retrieves entries modified by Anne Smith:

```
ldapsearch -h sun1 -b "" "(&(objectclass=*)(modifiersname=cn=Anne
Smith))"
```

The following example retrieves entries modified between 01 April 2001 and 06 April 2001:

```
ldapsearch -h sun1 -b "" "(&(objectclass=*)(modifytimestamp >= 20000401000000)
(modifytimestamp <= 20000406235959))"
```

Note: Because `modifiersname` and `modifytimestamp` are not indexed attributes, use catalog.sh to index these two attributes. Then, restart the Oracle directory server before issuing the two previous ldapsearch commands.

Other Examples: Each of the following examples searches on port 389 of host sun1, and searches the whole subtree starting from the DN `"ou=hr,o=acme,c=us"`.

The following example searches for all entries with any value for the objectclass attribute.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree "objectclass=*"
```

The following example searches for all entries that have `orcl` at the beginning of the value for the `objectclass` attribute.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"objectclass=orcl*"
```

The following example searches for entries where the `objectclass` attribute begins with `orcl` and `cn` begins with foo.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"(&(objectclass=orcl*)(cn=foo*))"
```

The following example searches for entries in which the common name (`cn`) is not `foo`.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree "(!(cn=foo))"
```

The following example searches for entries in which `cn` begins with `foo` or `sn` begins with `bar`.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"(|(cn=foo*)(sn=bar*))"
```

The following example searches for entries in which `employeenumber` is less than or equal to 10000.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"employeenumber<=10000"
```

ldapUploadAgentFile.sh Syntax

Use LdapUploadAgentFile.sh to load mapping and configuration information when you are synchronizing directories.

```
ldapUploadAgentFile.sh -name  <Profile Name>
-config < which configset the profile is associated to >
-LDAPhost  <LDAP Server host >
-LDAPport  <LDAP server port >
-binddn  < Dn that can modify the profile  ( default  = cn=orcladmin ) >
-bindpass <  password to the binddn ( default = welcome) >
-attrtype  <  "MAP" / "ATTR" >
-filename < Complete pathname of the file to be uploaded >
```

Table A–1 Arguments for ldapUploadAgentFile.sh

Argument	Description
Name	The Name of the Integration Profile to which the information needs to be loaded.
Config	The configset to which the Profile belongs to.
LDAPhost	LDAP Server host
LDAPport	LDAP Server Port
Binddn	Binddn of the directory user who has access rights to modify the profile entry.
Bindpass	Password corresponding to the binddn
AttrType	Type of file to be loaded. "MAP' is specified for loading the mapping file. And "ATTR" is specified for loading the config info file.
Filename	Complete pathname of the file to be uploaded.

ldapCreateConn.sh Syntax

You can create an integration profile by using the command-line tool ldapcreateConn.sh. This tool is in the following directory:

$*ORACLE_HOME*/ldap/admin/.

The following example creates an integration profile named "HRMS" in configuration set 2:

```
ldapcreateConn.sh
        -name agent_name>
        [ -type  <IMPORT | EXPORT > ] \
        [ -agentpwd < Agent Password> ] \
        [ -config <which configset to associate to > ] \
        [ -LDAPhost <LDAP server host> ]
        [ -LDAPport <LDAP server port> ] \
        [ -binddn SuperUserDN (default cn=orcladmin ) ] \
        [ -bindpass  Bindpassword (default=welcome) ] \
        [ -retry   <Max Retry Count on synchronization Errors > ] \
        [ -poll < Polling Interval For Synchronization> ] \
        [ -host < Host on which to run Agent> ] \
        [ -conndirurl  < Connected Directory URL > ] \
        [ -conndiracct < Connected Directory Acct Info > ] \
        [ -conndirpwd < Connected Directory Acc Pwd> ] \
        [ -execmd < Command Line for the Agent > ]  \
        [ -iftype < Interface Type > ]      \
        [ -condirfilter < Connected Directory Matching Filter> ]\
        [ -oidfilter < OID Matching Filter > ] \
        [ -U <SSL Authentication Mode> ]
        [ -W <Wallet location> ]\
        [ -P <Wallet password> ]
```

Table A–2 Arguments for Registering a Partner Agent by Using ldapcreateConn.sh

Argument	Description
Name	The name of the Integration Profile.This must be unique.
Type	IMPORT/EXPORT. The default is IMPORT/
Agentpwd	The password to protect the profile. The default is ‘welcome’.
Config	The configuration set number. The default is 1.
LDAPhost	The LDAP Server host. The default is the current host.
LDAPport	The LDAP server port The default is port 389.
Binddn	The BIND DN of the Directory user which has the privileges to create Integration profile. The default is ‘cn=orcladmin’
Bindpass	The BIND password. The default is ‘welcome’
Retry	Maximum number of retries to be done by the server on encountering a synchronization error. The default is ‘5’.

Table A–2 Arguments for Registering a Partner Agent by Using ldapcreateConn.sh

Argument	Description
Poll	The scheduling interval of the profile. The default is '60' seconds.
Host	This is currently used. For the time being, it should be set to the machine name on which the DIP server is executing.
Conndirurl	The Connected Directory access Information.
Conndiracct	The Connected Directory account.
Conndirpwd	The Connected Directory account password
Execmd	The OS command line to execute the partner agent.
Iftype	The Interface Type. The default is TAGGED.
Condirfilter	The Connected Directory Matching Filter
Oidfilter	The OID Matching Filter.

StopOdiServer.sh Syntax

In a client-only installation where the monitor and oidctl tools are not available, you can start the directory integration server without the oidctl tool. To stop the server, use the stopOdiServer.sh tool.

The path name for this tool is: $*ORACLE_HOME*/ldap/admin/stopodiserver.sh

The usage is:

```
$ORACLE_HOME/ldap/admin/stopodiserver.sh
        [ -LDAPhost LDAP_server_host ]
        [ -LDAPport LDAP_server_port ]
        [ -binddn super_user_dn (default cn=orcladmin ) ]
        [ -bindpass   bind_password (default=welcome) ]
        -instance instance_number_to_stop
```

Table A–3 Arguments for Stopping the

Argument	Description
LDAPhost	The LDAP Server host. The default is the current host.
LDAPport	The LDAP server port The default is port 389.

Table A–3 Arguments for Stopping the

Argument	Description
Binddn	The BIND DN of the Directory user which has the privileges to create Integration profile. The default is 'cn=orcladmin'
Bindpass	The BIND password. The default is 'welcome'
Instance	The instance number of the DIP server to stop.

Provisioning Subscription Tool Syntax

Use the Provisioning Subscription Tool to administer provisioning profile entries in the directory. More specifically, use it to perform these activities:

- Create a new provisioning profile. A new provisioning profile is created and set to the enabled state so that the Oracle Directory Integration Platform can process it
- Disable an existing provisioning profile
- Enabled a disabled provisioning profile
- Delete an existing provisioning profile
- Get the current status of a given provisioning profile
- Clear all of the errors in an existing provisioning profile

The Provisioning Subscription Tool shields the location and schema details of the provisioning profile entries from the callers of the tool. From the callers' perspective, the combination of an application and a subscriber uniquely identify a provisioning profile. The constraint in the system is that there can be only one provisioning profile per application per subscriber.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

The name of the executable is oidProvTool, located in `$ORACLE_HOME/bin`.

To invoke this tool, use this command:

```
oidprovtool param1=param1_value  param2=param2_value param3=param3_value ...
```

The Provisioning Subscription Tool accepts the following parameters:

Table A–4

Name	Operations	Mandatory/Optional	Description
operation	all	M	The subscription operation to be performed. The legal values for this parameter are: create, enable, disable, delete, status and reset. Only one operation can be performed per invocation of the tool.
ldap_host	all	O	Host-name of the LDAP server on which the subscription operations are to be performed. If not specified, the default value of 'localhost' is assumed.
ldap_port	all	O	The TCP/IP port on which the LDAP server is listening for requests. If not specified, the default value of '389' is assumed.
ldap_user_dn	all	M	The LDAP distinguished name of the user on whose behalf the operation is to be performed. Not all users have the necessary permissions to perform Provisioning Subscription operations. Please see the administrative guide to grant or deny LDAP users the permission to perform Provisioning Subscription operations.
ldap_user_password	all	M	The password of the user on whose behalf the operation is to be performed.
application_dn	all	M	The LDAP distinguished name of the application for which the Provisioning Subscription Operation is being performed. The combination of the application_dn and the organization_dn parameters help the subscription tool to uniquely identify a provisioning profile.

Table A–4

Name	Operations	Mandatory/Optional	Description
organization_dn	all	M	The LDAP distinguished name of the organization for which the Provisioning Subscription Operation is being performed. The combination of the application_dn and the organization_dn parameters help the subscription tool to uniquely identify a provisioning profile.
interface_name	create only	M	Database schema name for the PLSQL package. Format of the value should be: [Schema].[PACKAGE_NAME]
interface_type	create only	O	The type of the interface to which events have to be propagated.Valid Values: PLSQL (if not specified this is assumed as the default)
interface_connect_info	create only	M	Database connect string Format of this string:[HOST]:[PORT]:[SID]:[USER_ ID]:[PASSWORD]
interface_version	create only	O	The version of the interface protocol. Valid Values: 1.0 or 1.11.0 will be the old interface. If not specified, this is used as the default.
interface_additional_info	create only	O	Additional information for the interface. This is not currently used.

Table A–4

Name	Operations	Mandatory/Optional	Description
schedule	create only	O	The scheduling information for this profile. The value is the length of the time interval in seconds after which DIP will process this profile. If not specified, a default of 3600 is assumed.
max_retries	create only	O	The number of times the Provisioning Service should retry a failed event delivery. If not specified, a default value of 5 is assumed.
event_subscription	create only	O	Events for which DIP should send notification to this application. Format of this string:"[USER]GROUP]:[*Domain of interest*>]:[DELETE]ADD]MODIFY(<*list of attributes separated by comma*>)]"Multiple values may be specified by listing the parameter multiple times each with different values. If not specified the following defaults are assumed:USER:<org. DN>:DELETEGROUP:<org. DN>:DELETE(i.e. send user and group delete notifications under the organization DN).

Bulk Tools Syntax

This section contains these topics:

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Note: All bulk tools require you to enter the correct password in order to access the ods database.

bulkdelete Syntax

The bulkdelete command-line tool enables you to delete a subtree efficiently. It can be used when both an Oracle directory server and Oracle directory replication servers are in operation. It uses a SQL interface to benefit performance. For this release, the bulkdelete tool runs on only one node at a time.

This tool does not support filter-based deletion. That is, it deletes an entire subtree below the root of the subtree. If the base DN is a user-added DN, rather than a DN created as part of the installation of the directory, it is included in the delete. You must restrict LDAP activity against the subtree during deletion.

The bulkdelete tool uses this syntax:

```
bulkdelete.sh -connect net_service_name -base "base_dn" -size number_of_entries
-encode "character_set"
```

Mandatory Argument	Description
- connect *net_service_name*	Specifies the net service name to connect to the directory database **See Also:** *Oracle9i Net Services Administrator's Guide* in the Oracle Database Documentation Library
- base "*base_dn*"	Specifies the base DN of the subtree to be deleted

Optional Argument	Description
-size *number_of_entries*	Specifies the number of entries to be committed as a part of one transaction.
-encode "*character_set*"	Native character set encoding

bulkload Syntax

The bulkload command-line tool uses Oracle SQL*Loader to create directory entries from data residing in or created by other applications. When using bulkload, you specify any options and the input filename. Bulkload expects an empty directory and will either fail or overwrite if there are existing entries.The bulkload tool expects the input file to be in LDIF.

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

The bulkload tool uses this syntax:

```
bulkload.sh -connect net_service_name [-check] [-generate] [-load]
   [-restore] absolute_path_to_ldif.file
```

Mandatory Argument	Description
connect *net_service_name*	Specifies the net service name defined in the `tnsnames.ora` file. **See Also:** *Oracle9i Net Services Administrator's Guide* in the Oracle Database Documentation Library

Optional Argument	Description
-check	Checks LDAP schema for inconsistencies and for existence of duplicate DNs in the file
-encode "*character_set*"	Specifies native character set encoding. See Chapter 8, "Globalization Support in the Directory".
-generate	Creates files suitable for loading into Oracle Internet Directory
-load	Loads files resulting from generate phase into specified database
-restore	Takes the operational attributes, such as `orclguid`, `creatorsname`, and `createtimestamp`, from the LDIF file rather than generating new ones. Use this argument only when the LDIF file contains operational attributes. Use this in conjunction with the `generate` and `check` arguments.

Bulk loading must be performed when directory server instances are not running.

> **See Also:** Chapter 5, "Oracle Directory Server Administration" for instructions on stopping directory server instances

The LDIF data file path must be fully specified for check or generate operations.

Note: If bulkload.sh is not used to populate the directory, then $ORACLE_HOME/ldap/admin/oidstats.sh must be run to ensure there will be no significant search performance degradation.

Bulk Loading Multiple Nodes in a Replicated Environment

After generating a file with the `generate` option, you can use the `load` option to load multiple computers with the identical SQL*Loader file. Do this only when creating a new replica node.

> **See Also:** "Oracle Directory Replication Server Administration" on page 23-1

The current version of bulkload does not allow you to specify the connection information for all of the nodes in one command.

When you load the same data into multiple nodes in a replicated network, ensure that the `orclGUID` parameter (global IDs) is consistent across all the nodes. You can accomplish this by generating the bulkload data file once only (using the

-generate option), and then using the same data file to load the other nodes (using the -load option).

bulkmodify Syntax

The bulkmodify command-line tool enables you to modify a large number of existing entries in an efficient way. The bulkmodify tool supports the following:

- Subtree based modification
- A single attribute filter. For example, the filter could be objectclass=*, objectclass=oneclass, or telephonenumber=*.
- Attribute value addition and replacement. It modifies all matched entries in bulk.

The bulkmodify tool performs schema checking on the specified attribute name and value pair during initialization. All entries that meet the following criteria are modified:

- They are under the specified subtree.
- They meet the single filter condition.
- They contain the attribute to be modified as either mandatory or optional.

The Oracle directory server and Oracle directory replication server may be running concurrently while bulk modification is in progress, but the bulk modification does not affect the replication server. You must perform bulk modification against all replicas.

> **Note:** LDIF file based modification is not supported by bulkmodify. This type of modification requires per entry based schema checking, and therefore the performance gain over the existing ldapmodify tool is insignificant.

You must restrict user access to the subtree during bulk modification. If necessary, **ACI** restriction can be applied to the subtree being updated by bulkmodify.

You cannot use bulkmodify to add a value to single-valued attributes that already contain one value. If a second value is added, you must alter the directory schema to make that attribute multi-valued.

The bulkmodify tool uses this syntax:

```
bulkmodify -c net_service_name -b "base_dn" {-a|-r} attr_name -v att_value [-f
filter] [-s size]
```

Mandatory Argument	Description
-c *net_service_name*	Specifies the net service name of the directory database **See Also:** *Oracle9i Net Services Administrator's Guide* in the Oracle Database Documentation Library
-b "*base_dn*"	Specifies the base DN of the subtree to be modified
-a *attr_name*	Specifies the attribute name for addition
-r *attr_name*	Specifies the attribute name for replacement
-v *att_value*	Specifies the attribute value for either addition or replacement

Optional Argument	Description
-f *filter*	Specifies the filter to be used
-s *number_of_entries*	Specifies the number of entries to be committed as a part of one transaction. If not specified, default is 100.
-E "*character_set*"	Specifies native character set encoding. See Chapter 8, "Globalization Support in the Directory".

The filter specified with the `-f` option must contain a single attribute.

If a filter is not specified, the default filter `objectclass=*` is assumed.

There can be only one attribute name specified in the -a or `-r` option in each execution.

There can be only one value specified in the `-v` option in each execution. For example, the following bulkmodify command adds the telephone number 408-123-4567 to the entries of all employees who have Anne Smith as their manager:

```
bulkmodify -c my_database -b "c=US" -a telephoneNumber -v "408-123-4567" -f
"manager=Anne Smith"
```

To assure that the modified entries are read, after completing the bulkmodify procedure, restart the Oracle Internet Directory server.

ldifwrite Syntax

The ldifwrite command-line tool enables you to convert to LDIF all or part of the information residing in an Oracle Internet Directory. This makes that information available for loading into a new node in a replicated directory or into another node for backup storage.

> **Note:** The ldifwrite tool output does not include operational data of the directory itself—for example, `cn=subschemasubentry`, `cn=catalogs`, and `cn=changelog entries`. To export these entries into LDIF format, use ldapsearch with the `-L` flag.

The ldifwrite tool performs a subtree search, including all entries below the specified DN, including the DN itself.

The ldifwrite tool uses this syntax:

`ldifwrite -c` *`net_service_name`* `-b "`*`base_DN`*`" -f` *`filename`*

Mandatory Argument	Description
-c *net_service_name*	Specifies the net service name of the directory that is the source of the data, as defined in the `tnsnames.ora` file. **See Also:** *Oracle9i Net Services Administrator's Guide* in the Oracle Database Documentation Library
-b "*base_dn*"	Specifies the base of the subtree to be written out in LDIF format
-f *filename*	Specifies the name of the LDIF file to be created

Optional Argument	Description
-E "*character_set*"	Specifies native character set encoding. **See Also:** "Using Globalization Support with ldifwrite" on page 8-9

The following example writes all the entries under `ou=Europe, o=imc, c=us` into the `output1.ldi` file.

```
ldifwrite -c nldap -b "ou=Europe, o=imc, c=us" -f output1.ldi
```

All the arguments are mandatory.

The LDIF file and the intermediate file are always written to the current directory.

The ldifwrite tool includes the operational attributes of each entry in the directory, including `createtimestamp`, `creatorsname`, and `orclguid`.

When prompted for the OiD password, enter the password of the underlying ODS user. The default password is `ods`.

Catalog Management Tool Syntax

Oracle Internet Directory uses indexes to make attributes available for searches. When Oracle Internet Directory is installed, the `cn=catalogs` entry lists available attributes that can be used in a search. You can index only those attributes that have:

- An equality matching rule
- Matching rules supported by Oracle Internet Directory
- No more than 28 characters in their names

> **See Also:** "Matching Rules" on page C-10 for the matching rules supported by Oracle Internet Directory

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

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

The Catalog Management tool uses this syntax:

```
catalog.sh -connect net_service_name {add|delete} {-attr attr_name|-file
filename}
```

Mandatory Argument	Description
- connect *net_service_name*	Specifies the net service name to connect to the directory database **See Also:** *Oracle9i Net Services Administrator's Guide* in the Oracle Database Documentation Library

Optional Argument	Description
- add -attr *attr_name*	Indexes the specified attribute
- delete -attr *attr_name*	Drops the index from the specified attribute
- add -file *filename*	Indexes attributes (one per line) in the specified file
-delete -file *filename*	Drops the indexes from the attributes in the specified file

When you enter the `catalog.sh` command, the following message appears:

```
This tool can only be executed if you know the OiD user password.
Enter OiD password:
```

If you enter the correct password, the command is executed. If you give an incorrect password, the following message is displayed:

```
Cannot execute this tool
```

To effect the changes after running the Catalog Management tool, stop, then restart, the Oracle directory server.

> **See Also:** "OID Control Utility Syntax" on page A-43 and for instructions on starting and restarting directory servers. Note that OID Monitor must be running before you start a directory server. See "OID Monitor Syntax" on page A-42 for information about starting OID Monitor.

Note: Be careful not to use the catalog.sh `-delete` option to remove indexes on attributes unless you are absolutely sure that the indexes were not created by the base schema that was installed with Oracle Internet Directory. Removing indexes from base schema attributes can adversely impact the operation of Oracle Internet Directory.

OID Monitor Syntax

This section contains these topics:

- Starting the OID Monitor
- Stopping the OID Monitor

Starting the OID Monitor

To start the OID Monitor:

1. Set the following environment variable to the appropriate language setting. The default language set at installation is AMERICAN_AMERICA.

   `NLS_LANG=`*`APPROPRIATE_LANGUAGE`*`.UTF8`

2. At the system prompt, type:

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

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

For example:

```
oidmon connect=dbs1 sleep=10 start
```

Stopping the OID Monitor

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

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

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

For example:

```
oidmon connect=dbs1 stop
```

OID Control Utility Syntax

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

This section contains these topics:

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

Starting and Stopping an Oracle Directory Server Instance

Use the **OID Control Utility** to start and stop Oracle directory server instances.

Starting an Oracle Directory Server Instance

The syntax for starting an Oracle directory server instance is:

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

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

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

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

When starting and stopping an Oracle directory server instance, the server name and instance number are mandatory. All other arguments are optional.

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

Single quotes are mandatory around the flags.

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

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

Stopping an Oracle Directory Server Instance

At the system prompt, type:

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

For example:

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

Starting and Stopping an Oracle Directory Replication Server Instance

Use the OID Control Utility to start and stop Oracle directory replication server instances.

Starting an Oracle Directory Replication Server Instance

The syntax for starting the Oracle directory replication server is:

```
oidctl connect=net_service_name server=oidrepld instance=server_instance_number
[configset=configset_number] flags='-h hostname -p port_number
-d debug_level -z transaction_size' start
```

Argument	Description
`connect`	If you already have a `tnsnames.ora` file configured, then this is the name specified in that file, which is located in ORACLE_HOME/network/admin
`server`	Type of server to start (valid values are OIDLDAPD and OIDREPLD). This is not case-sensitive.
`instance`	Instance number of the server to start. Should be a number between 0 and 1000.
`configset`	Configset number used to start the server. This defaults to configset0 if not set. This should be a number between 0 and 1000.
`-p`	Specifies a port number during server instance startup. Default port if not set is 389.
`-d`	Specifies a debug level during replication server instance startup
`-h`	Specifies the host name on which the server runs. (Replication server only)
`-m [true|false]`	Turns conflict resolution on and off. The default is true (values = true and false). (Replication server only)
`-z`	Specifies the number of changes applied in each replication update cycle. If you do not specify this, the number is determined by the Oracle directory server sizelimit parameter, which has a default setting of 1024. You can configure this latter setting.
`start`	Starts the server specified in the *server* argument.

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

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

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

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

Single quotes are mandatory around the flags.

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

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

Stopping an Oracle Directory Replication Server Instance

At the system prompt, type:

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

For example:

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

Restarting Directory Server Instances

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

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

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

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

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

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

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

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

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

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

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

Troubleshooting Directory Server Instance Startup

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

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

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

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

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

OID Database Password Utility Syntax

The OID Database Password Utility syntax is:

```
oidpasswd [connect=net_service_name]
```

The OID Database Password Utility prompts you for the current password. Type the current password, then the new password, then a confirmation of the new password.

The OID Database Password Utility assumes by default that the password being changed is that of the local database (as defined by *ORACLE_HOME* and ORACLE_*SID*). If you are changing the password on a remote database, you must use the `connect=net_service_name` option.

For example:

```
$ oidpasswd
current password: ods
new password: newsupersecret
confirm password: newsupersecret
password set.
$
```

Note: User responses are not echoed to the screen.

Human Intervention Queue Manipulation Tool Syntax

The Human Intervention Queue Manipulation Tool enables you to move the changes from the human intervention queue to either the retry queue or the purge queue. Moving the change to the purge queue means that there are no further attempts to re-apply the changelog entry. Perform the following general steps to address changes in the human intervention queue:

1. Shutdown the Oracle directory replication server.
2. Analyze the replication log.
3. Use the Human Intervention Queue Manipulation Tool to move the changes to either the retry queue or the purge queue as described in the following sections.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Moving a Change from the Human Intervention Queue into the Retry Queue

To place a change back into the retry queue, use this syntax:

```
hiqretry.sh –connect net_service_name [-start change_number]
[-end change_number] [-equal change_number] -supplier supplier_node
```

The arguments are:

Argument	Description
`-connect net_service_name`	Connects to the database using the net service name defined in the `tnsnames.ora` file
`-start change_number`	Specifies the start change number for the retry operation. If you skip this option, then the command moves all the changes with change numbers less than or equal to the specified end change number back to the retry queue.
`-end change_number`	Specifies the end change number for the retry operation. If you skip this option, then the command moves all the changes with change numbers greater than or equal to the specified start change number back to the retry queue.
`-equal change_number`	Specifies the change number. The command moves the exact change conflict back to the retry queue. This option should not be present when `-start` or `-end` is used.
`-supplier supplier_node`	Specifies the supplier node where the changes originate

Moving a Change from the Human Intervention Queue into the Purge Queue

To place a change into the purge queue, use this syntax:

```
hiqpurge.sh –connect net_service_name [-start change_number] [-end change_
number] [-equal change_number] -supplier supplier_node
```

Arguments are:

Argument	Description
`-connect` *net_service_name*	Connects to the database using the net service name defined in the tnsnames.ora file
`-start` *change_number*	Specifies the start change number for the purge operation. If you skip this option, then the command moves all the changes with change numbers less or equal to the specified end change number back to the purge queue.
`-end` *change_number*	Specifies the end change number for the purge operation. If you skip this option, then the command moves all the changes with change numbers greater or equal to the specified start change number back to the purge queue.
`-equal` *change_number*	Specifies the change number of the change. The command moves the exact change conflict back to the purge queue. This option should not be present when `-start` or `-end` is used.
`-supplier` *supplier_node*	Specifies the supplier node where the changes originate

Note: When using hiqretry.sh or hiqpurge.sh, if you do not want all changes to be moved, then you must supply either the `-equal` flag, or a combination of the `-start` and `-end` flags.

Examples: Using the Human Intervention Queue Manipulation Tool

The following examples illustrate how to use the Human Intervention Queue Manipulation Tool.

Example: Retrying and Discarding Changes

Suppose that, after analyzing the replication log, you decide to do the following:

- Retry changes coming from the supplier node, ldap_rep1, with change numbers between 10324 to 10579
- Discard changes with change numbers between 10581 to 10623.

To do this, you issue these two commands:

```
hiqretry.sh –connect oiddb1 -start 10324 -end 10579 –supplier ldap_rep1
hiqpurge.sh –connect oiddb1 -start 10581 -end 10623 –supplier ldap_repl
```

The first command moves changes originating in ldap_rep1 with change numbers from 10324 to 10579 back to the retry queue. The second command deletes changes that originate in the supplier ldap_repl and that have change numbers from 10581 to 10623.

Example: Moving a Single Change from the Human Intervention Queue to the Retry Queue

The following command moves the change with change number equal to 10519 back to the retry queue.

```
hiqretry.sh –connect oiddb1 –equal 10519 –supplier ldap_repl
```

Example: Moving a Group of Changes from the Human Intervention Queue to the Retry Queue

The following command moves all the changes with change number greater or equal to 10324 back to the retry queue.

```
hiqretry.sh –connect oiddb1 –start 10324 –supplier ldap_repl
```

The following command moves all the changes with change numbers less than or equal to 10579 back to the retry queue.

```
hiqretry.sh –connect oiddb1 –end 10579 –supplier ldap_repl
```

Example: Moving All Changes from the Human Intervention Queue to the Retry Queue

The following command includes no options. It moves all changes that originate in the supplier ldap_repl from the human intervention queue to the retry queue.

```
hiqretry.sh –connect oiddb1 –supplier ldap_repl
```

OID Reconciliation Tool Syntax

When the Oracle directory replication server encounters inconsistent data, you can use the OID Reconciliation Tool to synchronize the entries on the consumer with those on the supplier. When you do this, perform the following general steps:

1. Set the supplier and the consumer to read-only mode.
2. Ensure that the supplier and the consumer are in tranquil state. If they are not in a tranquil state, then wait until they have finished updating.

3. Identify the inconsistent entries or subtree on the consumer.
4. Use the OID Reconciliation Tool to fix the inconsistent entries or subtree on the consumer.
5. Set the participating supplier and consumer back to read-write mode.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

Reconciling Inconsistent Data by Using the OID Reconciliation Tool

The OID Reconciliation Tool uses this syntax:

`oidreconcile -h` *`supplier_host`* `-c` *`consumer_host`* `[-P` *`supplier_port`*`] [-p` *`consumer_port`*`] [-s` *`scope`*`] -b "`*`basedn`*`" -W` *`supplier_password`* `-w` *`consumer_password`* `[-T` *`thread`*`]`

Argument	Description
-h *supplier_host*	Supplier host. This can be a computer name or IP address.
-c *consumer_host*	Consumer host. This can be a computer name or IP address.
-P *supplier_port*	Supplier TCP port. If you do not specify this option, then the tool connects to the default port (389).
-p *consumer_port*	Consumer TCP port. If you do not specify this option, then the tool connects to the default port (389).
-s *scope*	Reconcile scope: subtree
-b "*basedn*"	Specifies the distinguished name of the entry on which to perform reconciliation.
-W *supplier_password*	The password of `cn=orcladmin` of the supplier node
-w *consumer_password*	The password of `cn=orcladmin` of the consumer node
-T *thread*	Worker thread

How the OID Reconciliation Tool Works

When the OID Reconciliation Tool receives the specified DN, it compares the `orclGuid` of the parent DN on both the supplier and the consumer.

If the global identification (`orclGuid`) of both parents match, and the option `-s` *`subtree`* is set, then the OID Reconciliation Tool does the following:

1. Deletes all the entries in the subtree on the consumer node
2. Replaces them with entries from the supplier node

For example, the following command replaces the whole subtree starting from `"ou=hr,o=acme,c=us"` on the consumer with the equivalent subtree on the supplier:

```
oidreconcile -h supplier_host -P 389 -c consumer_host -p 389
-b "ou=hr,o=acme,c=us" -s subtree -W supplier_password -w consumer_password
```

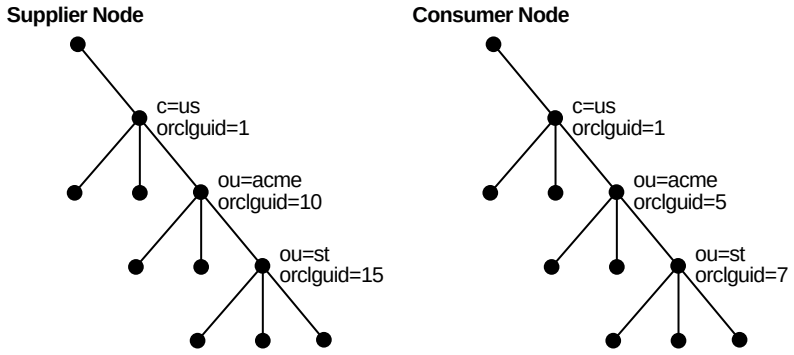
If the global identification (`orclGuid`) of both parents (`"o=acme,c=us"`) match, and `-s subtree` is not set, then the OID Reconciliation Tool replaces only the entry itself on the consumer node with the specified entry from the supplier node.

For example, the following command, in which the option `"-s subtree"` is not set, replaces only the specified entry, `"ou=hr,o=acme,c=us"`.

```
oidreconcile -h supplier -P 389 -c consumer -p 389 -b "ou=hr, o=acme, c=us"
-W supplier_password -w consumer_password
```

The next figure helps to explain how this process works.

Figure A–1 Example: OID Reconciliation Tool Process



This figure shows two DITs, one on a supplier node and one on a consumer node. In the DIT on the supplier node, the orclGuid for c=us is 1 (one), the orclGuid for o=acme is 10, and the orclGuid for ou=st is 15. On the consumer node, the orclGuid for o=acme is 5, and the orclGuid for ou=st is 7.

The orclGuids for the parent of o=acme,c=us—namely, c=us—on both the supplier and the consumer match. Therefore, the following command replaces all entries under o=acme,c=us on the consumer with the corresponding ones on supplier:

```
oidreconcile -h supplier -c consumer -b "o=acme, c=us" -s subtree -W supplier_
password -w consumer_password
```

If the orclGuid of both parents does not match, then the OID Reconciliation Tool does not perform the reconciliation. Instead, it tells the user the first ancestor on the consumer in which the orclGuid matches that of the same ancestor on the supplier.

For example, in the previous example, suppose you were to run the following command:

```
oidreconcile -h supplier -c consumer -b "ou=st, o=acme, c=us" -s subtree
-W supplier_password -w consumer_password
```

This command would result in a message that the first ancestor of ou=st in which the match of the orclGuid is o=acme,c=us. This message means that you should use o=acme,c=us as basedn argument for oidreconcile.

OID Database Statistics Collection Tool Syntax

Use the oidstats.sh tool to analyze the various database ods schema objects to estimate the statistics. It is located in the following directory: $*ORACLE_HOME*/ldap/admin/. The tool will prompt for 'ods' database user password.

Note: If you do not use the bulkload utility to populate the directory, then you must run the oidstats.sh tool to avoid significant search performance degradation.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

The OID Database Statistics Collection Tool uses this syntax:

```
oidstats.sh [ -connect net_service_name ]
            [ -all ]
            [ -cat catalog_name ]
            [ -pct percent ]
           [ -help | -usage ]
```

The parameters are:

Parameter	Description	Default
`connect` *`net_service_name`*	DB connect string	*ORACLE_SID*
`all`	Estimate statistics on all catalog tables plus DN catalogue	All catalogs
`cat` *`catalog_name`*	Estimate statistics either on all catalogs (all) or on a particular one, for example, ct_cn	None
`pct` *`percent`*	Percent of data to sample	100

Examples: Using the OID Database Statistics Collection Tool

Each of the following examples assume that the ORACLE_*SID* and the default user name and password are in effect.

The following example estimates statistics based on 100 percent sample data of all tables:

```
oidstats.sh -all -pct 100
```

The following example estimates statistics based on 50 percent sample data of all tables:

```
oidstats.sh -all -pct 50
```

The following example estimates statistics based on 50 percent sample data of CT_CN table:

```
oidstats.sh -cat ct_cn -pct 50
```

The following example estimates statistics based on 40 percent sample data of all catalog tables:

```
oidstats.sh -cat all -pct 40
```

SchemaSync Syntax

SchemaSync enables you to synchronize schema elements—namely attributes and object classes—between an Oracle directory server and third-party LDAP directories.

The usage for SchemaSync is as follows:

```
$ORACLE_HOME/bin/schemasync
        -srchost source_LDAP_directory
        -srcport <source_LDAP_port_numbert -srcdn privileged_DN_in_source_
        directory_to_access_schema
        -srcpwd password
        -dsthost destination_LDAP_directory
        -dstport destination_LDAP_port
        -dstdn privileged_dn_in_destination_directory_to_access_schema
        -dstpwd password
        [-ldap]
```

Note: `-ldap` is an optional parameter. If it is specified, then the schema changes are applied directly from the source LDAP directory to the destination LDAP directory. If it is not specified, then the schema changes are placed in the following LDIF files:

- `$ORACLE_HOME/ldap/odi/data/attributetypes.ldif`
  This file has the new attribute definitions.
- `$ORACLE_HOME/ldap/odi/data/objectclasses.ldif`
  This file has the new object class definitions.

if you do not specify `-ldap`, then you must use ldapmodify to upload the definitions from these two files, first attribute types and then object classes.

The errors that occur during schema synchronization are logged in the following log files:

- $*ORACLE_HOME*/ldap/odi/log/attributetypes.log
- $*ORACLE_HOME*/ldap/odi/log/objectclasses.log

B

The Access Control Directive Format

This appendix describes the format (syntax) of any **access control item (ACI)**. It contains these topics:

- Schema for orclACI
- Schema for orclEntryLevelACI

Schema for orclACI

The access control directive defined by the user attribute `orclACI` has the following schema:

```
OrclACI:
{ object_identifier NAME 'orclACI' DESC 'Stores an inheritable ACI' EQUALITY
accessDirectiveMatch SYNTAX 'accessDirectiveDescription'  USAGE
'directoryOperation'}
```

accessDirectiveDescription has the following BNF:

```
<accessDirectiveDescription>
                  ::= access to <object> [by <subject> ( <accessList> )]+

<object> ::= [attr <EQ-OR-NEQ> ( * | (<attrList>) ) | entry]
[filter=(<ldapFilter>)]

<subject> ::= <entity> [<BindMode>] [Added-object-constraint=(<ldapFilter>)]

<entity> ::= * | self | dn="<regex>" | dnAttr=(<dn_attribute>) | group="<dn>" |
guidattr=(<guid_attribute>) | groupattr=(<group_attribute>)

<BindMode> ::= | BindMode = Simple
               | BindMode = SSLNoauth
               | BindMode = SSLOneway
               | BindMode = SSLTwoway

<accessList> ::= <access> | <access>, <accessList>

<access> ::= none | compare | search | browse | proxy | read | selfwrite | write
| add | delete | nocompare | nosearch | nobrowse | noproxy |noread | noselfwrite
| nowrite | noadd | nodelete

<attrList> ::=  <attribute name> | <attribute name>,<attrList>

<EQ-OR-NEQ> ::=  = | !=

<regex> ::= <dn> | *,<dn_of_any_subtree_root>
```

Note: The regular expression defined above is not meant to match any arbitrary expression. The syntax only allows expressions where the wild card is followed by a comma and a valid DN. The latter DN denoted by <*dn_of_any_subtree_root*> is intended to specify the root of some subtree.

Schema for orclEntryLevelACI

The entry level access control directive defined by the user attribute `orclEntryLevelACI` has the following schema:

```
"orclEntryLevelACI":
{ object_identifier NAME 'orclEntryLevelACI' DESC 'Stores entry level ACL
Directive'
EQUALITY accessDirectiveMatch SYNTAX 'orclEntryLevelACIDescription'
USAGE 'directoryOperation' }


<orclEntryLevelACIDescription>
::= access to <object> [by <subject> ( <accessList> )]+
```

C

Schema Elements

This appendix briefly lists different schema elements supported by Oracle Internet Directory. Most of these elements are used as defined by the ldapext and ASID working groups of the Internet Engineering Task Force (IETF).

See Also: The following URLs on the World Wide Web:

- `http://www.ietf.org` for the IETF home page
- `http://www.ietf.org/html.charters/ldapext-charter.html` for the ldapext charter and LDAP drafts)
- `http://www.ietf.org/html.charters/ldup-charter.html` for the LDUP charter and drafts
- `http://www.iana.org`, the Internet Assigned Numbers Authority home page, for information about object identifiers

This appendix contains these topics:

IETF Requests for Comments (RFCs) Enforced by Oracle Internet Directory

Oracle Internet Directory enforces the following Requests for Comments (RFCs) of the Internet Engineering Task Force (IETF):

RFC	Title	URL
1777	Lightweight Directory Access Protocol	http://www.ietf.org/rfc/rfc1777.txt
1778	The String Representation of Standard Attribute Syntaxes	http://www.ietf.org/rfc/rfc1778.txt
1779	A String Representation of Distinguished Names	http://www.ietf.org/rfc/rfc1779.txt
1960	A String Representation of LDAP Search Filters	http://www.ietf.org/rfc/rfc1960
2079	Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs)	http://www.ietf.org/rfc/rfc2079.txt
2247	Using Domains in LDAP/X.500 Distinguished Names	http://www.ietf.org/rfc/rfc2247.txt
2251	Lightweight Directory Access Protocol (v3)	http://www.ietf.org/rfc/rfc2251.txt
2252	Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions	http://www.ietf.org/rfc/rfc2252.txt
2253	Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names	http://www.ietf.org/rfc/rfc2253.txt
2254	The String Representation of LDAP Search Filters	http://www.ietf.org/rfc/rfc2254.txt
2255	The LDAP URL Format	http://www.ietf.org/rfc/rfc2255.txt
2256	A Summary of the X.500(96) User Schema for use with LDAPv3	http://www.ietf.org/rfc/rfc2256.txt

IETF Drafts Enforced by Oracle Internet Directory

Oracle Internet Directory enforces the following two drafts of the IETF:

Draft Title	URL
"Definition of the inetOrgPerson LDAP Object Class"	`http://ietf.org/rfc/rfc2798.txt`
"Referrals and Knowledge References in LDAP Directories"	`http://www.ietf.org/proceedings/99nov/I-D/draft-ietf-ldapext-knowledge-00.txt`

Proprietary Oracle Internet Directory Schema Elements

Oracle Internet Directory's proprietary schema includes attributes and object classes in these categories:

- Access Control
- Replication
- Oracle Internet Directory Configuration
- SSL
- Audit Log
- Configuration Set Entry Attributes

In addition, Oracle Internet Directory installation includes schema elements that enable specific Oracle products to use Oracle Internet Directory. For information about these schema elements, see the documentation for the specific Oracle product.

Access Control

Attributes	Object Class
`orclEntryLevelACI, orclACI`	`orclPrivilegeGroup`

Replication

Attributes	Object Classes
orclGUID, changeNumber changeType, changes, orclParentGUID, server, supplier, consumer, orclReplBindDN, orclReplBindPassword, changeLog, changeStatus, orclChangeRetryCount, orclPurgeSchedule, orclDirReplGroupAgreement, orclAgreementId, orclSupplierReference, orclConsumerReference, orclReplicationProtocol, orclUpdateSchedule, targetDN, orclExcludedNamingcontexts, orclDirReplGroupDSAs	changeLogEntry, changeStatusEntry, orclReplAgreementEntry

Oracle Internet Directory Configuration

Attributes	Object Class
orcldebugflag, orclMaxCC, orclDBType, orclSuffix, orclDITRoot, orclSuName, orclSuPassword, orclSizeLimit, orclTimeLimit, orclGuName, orclGuPassword, orclServerProcs, orclconfigsetnumber, orclhostname, orclIndexedAttribute, orclCatalogEntryDN, orclServerMode, orclPrName, orclPrPassword, orclUseEncrypt, orclDirectoryVersion	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	Object Class
`orclServerEvent, orcleventtype, orclauditattribute, orclauditmessage, orcleventtime, orcluserdn, orclSequence, orclAuditLevel, orclOpResult`	`OrclAuditOC`

Configuration Set Entry Attributes

The following table lists and describes the entire set of configuration set entry attributes that are used to configure an instance of a directory server.

Parameter	Description
`orcldebugflag`	Debug level associated with this instance of the server. The default for configset0 is 0. The range is 0 to 65535.
`orclmaxcc`	Maximum number of concurrent database connections. The default for configset0 is 10. You cannot use a negative value for this attribute.
`orclserverprocs`	Number of server processes to start. The default for configset0 is 1. You cannot use a negative value for this attribute.

Parameter	Description
orclsslport	SSL mode default port (default 636). When you run the directory in the secure mode, it listens at default port 636 and accepts only SSL-based TCP/IP connections. (When you run the directory in the normal mode, it listens at default port 389, accepting normal TCP/IP connections.) You might want to change this port when you add multiple LDAP server instances.
orclnonsslport	Non-SSL mode default port (default 389).
orclsslenable	Flag for toggling SSL on and off. You would want to toggle this flag when you use different instances of the same server for either SSL or non-SSL. You may use either of the following two values: ■ 0 = disables SSL (default in configuration set0) ■ 1 = enables SSL The default is 0.
orclsslauthentication	Flag, with values of 1, 32, or 64, for specifying the type of authentication you elect to use for each instance of the Oracle directory server. The default value, 1, specifies no authentication. You can run different values concurrently for different instances. Values of one-way and two-way authentication require wallets. You may use one of the following three values: ■ 1 = no SSL authentication ■ 32 = one-way SSL authentication (the server sends its certificate to the client) ■ 64 = two-way SSL authentication (client and server send certificates to each other)

orclsslwalleturl	Sets the location of the Oracle wallet. You initially set this value when you create the wallet. If you elect to change the location of the Oracle wallet, you must change this parameter. You must set the wallet location on both the client and the server. For example, on Solaris, you could set this parameter as follows:
	`orclsslwalleturl=file:/Home/my_dir/`
	On Windows NT, you could set this parameter as follows:
	`file:Home\my_dir\`

Parameter	Description
`orclsslwalletpasswd`	Password used by the server to open its wallet. You initially set this value when you create the wallet. If you elect to change the wallet password, you must change this parameter. You must set the wallet password on both the client and the server.
`orclsslversion`	SSL version. The default is 3.

See Also:

- "Setting Debug Logging Levels by Using the OID Control Utility" on page 5-27 for information on debug levels
- Appendix D, "Oracle Wallet Manager" for information on setting the location of the Oracle Wallet and the Oracle Wallet password

LDAP Syntax

Syntax defines the type of values that an attribute can hold. Oracle Internet Directory recognizes most of the syntax specified in RFC 2252, that is, it allows you to associate most of the syntax described in that document with an attribute. In addition to recognizing most LDAP syntax, Oracle Internet Directory enforces some LDAP syntax.

This section covers topics in the following subsections:

- LDAP Syntax Enforced by Oracle Internet Directory
- Commonly Used LDAP Syntax Recognized by Oracle Internet Directory
- Additional LDAP Syntax Recognized by Oracle Internet Directory
- Size of Attribute Values

LDAP Syntax Enforced by Oracle Internet Directory

Oracle Internet Directory enforces LDAP syntax for the following:

- DN
- Facsimile Telephone Number
- OID (object identifier)

- Telephone Number

Note: The values you specify for these attributes must conform to the syntax specified in RFC 2252.

Commonly Used LDAP Syntax Recognized by Oracle Internet Directory

The following LDAP syntax is more commonly used:

Attribute Type Description	Presentation Address
Numeric String	Facsimile Telephone Number
Boolean	Printable String
Object Class Description	INTEGER
Certificate	Telephone Number
Octet String	JPEG
Directory String	UTC Time
OID	Name And Optional UID
DN	

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

LDAP Schema Description

ACI Item

LDAP Syntax Description

Audio

Mail Preference

Binary

Master And Shadow Access Points

Bit String

Matching Rule

Certificate List

Matching Rule Use Description

Certificate Pair

MHS OR Address

Country String

Modify Rights

Data Quality Syntax

Name Form Description

Delivery Method

Object Class Description

DIT Content Rule Description

Octet String

DIT Structure Rule Description

Other Mailbox

DL Submit Permission

Postal Address

DSA Quality Syntax

Protocol Information

DSE Type

Substring Assertion

Enhanced Guide

Subtree Specification

Fax

Supplier And Consumer

Generalized Time

Supplier Information

Guide

Supplier Or Consumer

IA5 String

Supported Algorithm

LDAP Schema Definition

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

IntegerMatch

bitStringMatch

numericStringMatch

caseExactMatch

objectIdentifierFirstComponentMatch

caseExactIA5Match

ObjectIdentifierMatch

caseIgnoreIA5Match

OctetStringMatch

caseIgnoreListMatch

presentationAddressMatch

caseIgnoreMatch

protocolInformationMatch

caseIgnoreOrderingMatch

telephoneNumberMatch

distinguishedNameMatch

uniqueMemberMatch

generalizedTimeMatch

generalizedTimeOrderingMatch

Of the matching rules in the previous list, Oracle Internet Directory actually enforces the following when it compares attribute values:

distinguishedNameMatch

caseExactMatch

caseIgnoreMatch

numericStringMatch

IntegerMatch

telephoneNumberMatch

Schema to Represent a User

A user is represented using the following object classes: OrclUser, OrclUserV2, in addition to inetOrgPerson. The following table describes the attribute names.

Attribute Name	Mandatory or Optional	Description
OrclGUID	Optional	Specifies a Unique Global ID to identify the user.
Cn	Mandatory	Specifies user's first name and / or common nickname.
Sn	Mandatory	Specifies a user's last name or surname.
GivenName	Optional	Specifies a user's given name.
MiddleName	Optional	Specifies a user's middle name, if any.
DisplayName	Optional	Specifies the name used by GUI tools for display purposes.
OrclMaidenName	Optional	Specifies a user's maiden name, if any.
OrclDateOfBirth	Optional	Specifies a user's birthdate, includes year in yyyymmdd format.
Street	Optional	Specifies the street and location associated with a user's office address.
L	Optional	Specifies the city for a user's office address.
PostalCode	Optional	Specifies the postal code associated with a user's office address.
St	Optional	Specifies the state associated with a user's office address.
C	Optional	Specifies the country associated with a user's office address.
EmployeeNumber	Optional	Specifies a user's employee number, if applicable.
O	Optional	Specifies the organization for which a user works.
Title	Optional	Specifies a user's designation.
Manager	Optional	Specifies the DN of a user's manager.
OrclHireDate	Optional	Specifies the date on which a user was hired by the organization.
Mail	Optional	Specifies a user's e-mail address.
JpegPhoto	Optional	Specifies a photograph of a user.

Attribute Name	Mandatory or Optional	Description
TelephoneNumber	Optional	Specifies a user's office or work telephone number.
Mobile	Optional	Specifies a user's mobile phone number.
Pager	Optional	Specifies a user's pager number.
FacsimileTelephoneNumber	Optional	Specifies a user fax number.
HomePostalAddress	Optional	Specifies the complete residential postal address of a user. The value is specified as $ separated values for different address components. For example, XYZ Avenue $ Apt. 2 $ San Francisco $ CA $ 92345 $ USA
HomePhone	Optional	Specifies a user's residential phone number.
UserPassword	Optional	Specifies a password to be used for authenticating a user.
OrclActiveStartDate	Optional	Specifies the time from which the user should be allowed to authenticate. The Value is represented in Universal Coordinated Time (UTC) time format. If the attribute is missing, the user is allowed to authenticate immediately.
OrclActiveEndDate	Optional	Specifies the date beyond which a user should not be allowed to authenticate. The value is represented in UTC time format.
OrclPasswordHint	Optional	Specifies the hint to use if a user forgets their password.
OrclPasswordHintAnswer	Optional	Specifies the answer to the password hint question.
OrclIsEnabled	Optional	Specifies if a user is currently enabled to authenticate. Valid values are ENABLED (or attribute not present in the user entry) and DISABLED. A user can successfully authenticate only if a user is enabled or the attribute is not present in the entry.
PreferredLanguage	Optional	Specifies the preferred language for communication with a user.

Attribute Name	Mandatory or Optional	Description
OrclTimeZone	Optional	Specifies the time zone applicable for a user location.
OrclDefaultProfileGroup	Optional	Specifies the DN of the group to use as default for a user's profile.
OrclIsVisible	Optional	Specifies if a user should display in a regular user search. Valid values are TRUE (or not present) and FALSE. If the attribute is not present, then a user record is visible.
OrclDisplayPersonal Information	Optional	Specifies if a user chooses to display personal information in a user search. Valid values are TRUE (or not present) and FALSE.
OrclWorkflowNotification Preference	Optional	Specifies the preferred delivery mechanism for sending workflow notification to a user.

D

Oracle Wallet Manager

Security administrators use Oracle Wallet Manager to manage public-key security credentials on Oracle clients and servers. The wallets it creates are opened by using either the Oracle Enterprise Login Assistant or the Oracle Wallet Manager.

This chapter describes the Oracle Wallet Manager, in the following sections:

- Overview
- Managing Wallets
- Managing Certificates

> **See Also:** *Oracle Advanced Security Administrator's Guide* in the Oracle Database Documentation Library for information about how to open and close wallets for secure SSL communications by using Oracle Enterprise Login Assistant

Overview

Traditional private-key or symmetric-key cryptography requires that entities desiring to establish secure communications possess a single secret key known only to them. *Harriet* and *Dick*, for example, could agree to shift each letter in their private messages by two character positions (A becomes C, B becomes E, and so on) to encrypt the message text. Using this method, a *HELLO* message from Harriet to Dick would read *JGNNP*. The actual encryption methods in current use are much more complex and significantly more secure, but an underlying problem remains—sending messages encrypted with a single key requires prior, *secure* distribution of the key to each participating party. Otherwise, a malicious third party might obtain the key, intercept communications, and compromise security. Public-key cryptography addresses this problem, by providing a secure method for key distribution.

Public-key cryptography requires a party to possess a **public/private key pair**. The **private key** is kept secret and is known only to that party. The **public key**, as the name implies, is freely available. To send a secret message to this party requires that a third party sender encrypt the message with the public key. Such a message can only be decrypted by a party holding the associated private key.

For example, when Dick wants to send a secure message to Harriet, he first asks Harriet for her public key (or obtains it from another, public source). Harriet gives Dick the public key, but Tom, a malicious eavesdropper, also obtains the public key. Nevertheless, when Dick sends Harriet a message encrypted with her public key, Tom cannot decrypt it; the message can only be decrypted with Harriet's private key.

Public-key algorithms thus guarantee the secrecy of a message, but they don't guarantee *secure communications* because they don't verify the identities of the communicating parties. In order to establish secure communications, it is important to verify that the public key used to encrypt a message does in fact belong to the target recipient. Otherwise, a third party can potentially eavesdrop on the communication and intercept public key requests, substituting its public key for a legitimate key.

If Tom, for example, is able to substitute his public key for Harriet's public key and send it to Dick, Dick might then send a message to Harriet encrypted with Tom's public key—believing he was using Harriet's public key. Tom could then decrypt a subsequent intercepted message from Dick using his private key, re-encrypt it with Harriet's public key and re-transmit it to Harriet. Harriet could then decrypt the incoming message using her private key, and never know that it had been intercepted by Tom.

In order to avoid such a man-in-the-middle attack, it is necessary to verify the owner of the public key, a process called **authentication**. This authentication can be accomplished through a **certificate authority (CA)**.

A CA is a third party that is trusted by both of the parties attempting secure communication. The CA issues public key certificates that contain an entity's name, public key, and certain other security credentials. Such credentials typically include the CA name, the CA signature, and the certificate effective dates (From Date, To Date).

The CA uses its private key to encrypt a message, while the public key is used to decrypt it, thus verifying that the message was encrypted by the CA. The CA public key is well known, and does not have to be authenticated each time it is accessed. Such CA public keys are stored in a **wallet**.

Oracle Wallet Manager is a stand-alone Java application that wallet owners use to manage and edit the security credentials in their Oracle wallets. These tasks include the following:

- Generating a public/private key pair and creating a certificate request for submission to a CA.
- Installing a certificate for the entity.
- Configuring one or more **trusted certificate** for the entity.
- Opening a wallet to enable access to PKI-based services.
- Creating a wallet that can be accessed by using either Oracle Enterprise Login Assistant or Oracle Wallet Manager.

Managing Wallets

This section describes how to create a new wallet and perform associated wallet management tasks, such as generating certificate requests, exporting certificate requests, and importing certificates into wallets, in the following subsections:

Starting Oracle Wallet Manager

To start Oracle Wallet Manager:

- On UNIX:

  Enter `owm` at the command line.

- On Windows NT:

  Press Start > *ORACLE_HOME* > Network Administration > Wallet Manager

Creating a New Wallet

Create a new wallet as follows:

1. Choose `Wallet > New` from the menu bar; the New Wallet dialog box appears.
2. Read the recommended guidelines for creating a password and enter a password in the Wallet Password field.

Because an Oracle wallet contains a user's credentials that can be used to authenticate the user to multiple databases, it is especially important to choose a strong password for the wallet. A malicious user who guesses the password to a user's wallet can access all the databases that the user can access.

Oracle Corporation recommends that you choose a password that is not too short, not easily guessed, and is reasonably complex. A reasonably complex password has at least six characters, and contains at least one symbol or number—so that it will not be found in a dictionary.

Example: `gol8fer`

It is also a prudent security practice for users to change their passwords periodically, such as once a month, or once a quarter.

3. Re-enter that password in the Confirm Password field.
4. Choose `OK` to continue.
5. An Alert is displayed, and informs you that a new empty wallet has been created. It prompts you to decide whether you want to create a certificate request. See: "Creating a Certificate Request" on page D-9.

   If you choose Cancel, you are returned to the Oracle Wallet Manager main window. The new wallet you just created appears in the left window pane. The certificate has a status of `Empty`, and the wallet displays its default trusted certificates.
6. Select `Wallet > Save In System Default` to save the new wallet.

   If you do not have permission to save the wallet in the system default, you can save it to another location.

   A message at the bottom of the window informs you that the wallet was successfully saved.

Opening an Existing Wallet

Open a wallet that already exists in the file system directory as follows:

1. Choose `Wallet > Open` from the menu bar; the Select Directory dialog box appears.
2. Navigate to the directory location in which the wallet is located, and select the directory.
3. Choose OK; the Open Wallet dialog box appears.

4. Enter the wallet password in the Wallet Password field.
5. Choose OK.
6. The message `Wallet opened successfully` appears at the bottom of the window, and you are returned to the Oracle Wallet Manager main window. The wallet's certificate and its trusted certificates are displayed in the left window pane.

Closing a Wallet

To close an open wallet in the currently selected directory:

- Choose `Wallet > Close`.
- The message `Wallet closed successfully` appears at the bottom of the window, to confirm that the wallet is closed.

Saving Changes

To save your changes to the current open wallet:

- Choose `Wallet > Save`.
- A message at the bottom of the window confirms that the wallet changes were successfully saved to the wallet in the selected directory location.

Saving the Open Wallet to a New Location

Use the `Save As` option to save the current open wallet to a new directory location:

1. Choose `Wallet > Save As`. The select directory dialog box appears.
2. Select a directory location to save the wallet.
3. Choose OK.

   The following message appears if a wallet already exists in the selected directory:

   ```
   A wallet already exists in the selected path. Do you
   want to overwrite it?.
   ```

   Choose Yes to overwrite the existing wallet, or No to save the wallet to another directory.

   A message at the bottom of the window confirms that the wallet was successfully saved to the selected directory location.

Saving in System Default

Use the `Save in System Default` menu option to save the current open wallet to the system default directory location. This makes the current open wallet the wallet that is used by SSL:

- Choose `Wallet > Save in System Default`.
- A message at the bottom of the window confirms that the wallet was successfully saved in the system default wallet location.

Deleting the Wallet

To delete the current open wallet:

1. Choose `Wallet > Delete`; the `Delete Wallet` dialog box appears.
2. Review the displayed wallet location to verify you are deleting the correct wallet.
3. Enter the wallet password.
4. Choose `OK`; a dialog panel appears to inform you that the wallet was successfully deleted.

Note: Any open wallet in application memory will remain in memory until the application exits. Therefore, deleting a wallet that is currently in use does not immediately affect system operation.

Changing the Password

A password change is effective immediately. The wallet is saved to the currently selected directory, with the new encrypted password.To change the password for the current open wallet:

1. Choose `Wallet > Change Password;` the `Change Wallet Password` dialog box appears.
2. Enter the existing wallet password.
3. Enter the new password.
4. Re-enter the new password.
5. Choose `OK`.

A message at the bottom of the window confirms that the password was successfully changed.

Using Auto Login

The Oracle Wallet Manager Auto Login feature opens a copy of the wallet and enables PKI-based access to secure services—as long as the wallet in the specified directory remains open in memory.

You must enable Auto Login if you want single sign-on access to multiple Oracle databases.

Enabling Auto Login

To enable Auto Login:

1. Choose `Wallet` from the menu bar.
2. Choose the check box next to the Auto Login menu item; a message at the bottom of the window displays `Autologin enabled`.

Disabling Auto Login

To disable Auto Login:

1. Choose `Wallet` from the menu bar.
2. Choose the check box next to the Auto Login menu item; a message at the bottom of the window displays `Autologin disabled`.

Using Oracle Wallet Manager with Oracle Application Server

When using the Oracle Application Server (OAS), you must install the Oracle Wallet Manager on a primary node and on each remote node in a multi-node configuration. After you install the product on each node you must then copy the wallet from the primary node to each of the remote nodes.

Managing Certificates

Oracle Wallet Manager uses two kinds of certificates: user certificates and trusted certificates. This section describes how to manage both certificate types, in the following subsections:

- Managing User Certificates
- Managing Trusted Certificates

> **Note:** You must first install a trusted certificate from the certificate authority before you can install a user certificate issued by that authority. Several trusted certificates are installed by default when you create a new wallet.

Managing User Certificates

Managing user certificates involves the following tasks:

- Creating a Certificate Request
- Exporting a User Certificate Request
- Importing the User Certificate into the Wallet
- Removing a User Certificate from a Wallet

Creating a Certificate Request

The actual certificate request becomes part of the wallet. You can reuse any certificate request to obtain a new certificate. However, you cannot edit an existing certificate request; store only a correctly filled out certificate request in a wallet.

To create a PKCS #10 certificate request:

1. Choose `Operations > Create Certificate Request`; the `Create Certificate Request` dialog box appears.
2. Enter the following information (Table D–1):

Table D–1 Certificate Request: Fields and Descriptions

Field Name	Description
Common Name	Mandatory. Enter the name of the user's or service's identity. Enter a user's name in first name / last name format.

Table D–1 Certificate Request: Fields and Descriptions

Field Name	Description
Organizational Unit	Optional. Enter the name of the identity's organizational unit. Example: Finance.
Organization	Optional.Enter the name of the identity's organization. Example: XYZ Corp.
Locality / City	Optional. Enter the name of the locality or city in which the identity resides.
State / Province	Optional. Enter the full name of the state or province in which the identity resides. Enter the full state name, because some certificate authorities do not accept two–letter abbreviations.
Country	Mandatory. Choose the drop-down list to view a list of country abbreviations. Select the country in which the organization is located.
Key Size	Mandatory. Choose the drop-down box to view a list of key sizes to use when creating the public / private key pair.
Advanced	Optional. Choose Advanced to view the Advanced Certificate Request dialog panel. Use this field to edit or customize the identity's distinguished name (DN). For example, you can edit the full state name and locality.

3. Choose OK. An Oracle Wallet Manager dialog box informs you that a certificate request was successfully created. You can either copy the certificate request text from the body of this dialog panel and paste it into an e-mail message to send to a certificate authority, or you can export the certificate request to a file.
4. Choose OK. You are returned to the Oracle Wallet Manager main window; the status of the certificate is changed to Requested.

Exporting a User Certificate Request

Save the certificate request in a file system directory when you elect to export a certificate request:

1. Choose `Operations > Export Certificate Request` from the menu bar; the Export Certificate Request dialog box appears.
2. Enter the file system directory in which you want o save your certificate request, or navigate to the directory structure under Folders.
3. Enter a file name to save your certificate request, in the Enter File Name field.
4. Choose `OK`. A message at the bottom of the window confirms that the certificate request was successfully exported to the file. You are returned to the Oracle Wallet Manager main window.

Importing the User Certificate into the Wallet

You will receive an e-mail notification from the certificate authority informing you that your certificate request has been fulfilled. Import the certificate into a wallet in either of two ways: copy and paste the certificate from the e-mail you receive from the certificate authority, or import the user certificate from a file.

Pasting the Certificate

To paste the certificate:

1. Copy the certificate text from the e-mail message or file you receive from the certificate authority. Include the lines `Begin Certificate` and `End Certificate.`
2. Choose `Operations > Import User Certificate` from the menu bar; the Import Certificate dialog box appears.
3. Choose the `Paste the Certificate` button, and choose `OK`; an Import Certificate dialog box appears with the following message:

   ```
   Please provide a base64 format certificate and paste it below.
   ```

4. Paste the certificate into the dialog box, and choose `OK`. A message at the bottom of the window confirms that the certificate was successfully installed. You are returned to the Oracle Wallet Manager main panel, and the wallet status changes to `Ready`.

Selecting a File that Contains the Certificate

To select the file:

1. Choose `Operations > Import User Certificate` from the menu bar.
2. Choose the `Select a file...` certificate button, and choose `OK`; the Import Certificate dialog box appears.
3. Enter the path or folder name of the certificate location.
4. Select the name of the certificate file (for example, `cert.txt`).
5. Choose `OK`. A message at the bottom of the window appears, to inform you that the certificate was successfully installed. You are returned to the Oracle Wallet Manager main panel, and the wallet status is changes to `Ready`.

Removing a User Certificate from a Wallet

1. Choose `Operations > Remove User Certificate`; a dialog panel appears and prompts you to verify that you want to remove the user certificate from the wallet.
2. Choose `Yes`; you are returned to the Oracle Wallet Manager main panel, and the certificate displays a status of `Requested`.

Managing Trusted Certificates

Managing trusted certificates includes the following tasks:

- Importing a Trusted Certificate
- Removing a Trusted Certificate
- Exporting a Trusted Certificate
- Exporting All Trusted Certificates
- Exporting a Wallet

Importing a Trusted Certificate

You can import a trusted certificate into a wallet in either of two ways: paste the trusted certificate from an e-mail that you receive from the certificate authority, or import the trusted certificate from a file.

Oracle Wallet Manager automatically installs trusted certificates from VeriSign, RSA, and GTE CyberTrust Entrust when you create a new wallet.

Pasting the Trusted Certificate To paste the trusted certificate:

1. Choose `Operations > Import Trusted Certificate` from the menu bar; the Import Trusted Certificate dialog panel appears.
2. Choose the `Paste the Certificate` button, and choose `OK`. An Import Trusted Certificate dialog panel appears with the following message:

   ```
   Please provide a base64 format certificate and paste it below.
   ```

3. Copy the trusted certificate from the body of the e-mail message you received that contained the user certificate. Include the lines `Begin Certificate` and `End Certificate.`
4. Paste the certificate into the window, and Choose `OK`. A message at the bottom of the window informs you that the trusted certificate was successfully installed.
5. Choose `OK`; you are returned to the Oracle Wallet Manager main panel, and the trusted certificate appears at the bottom of the Trusted Certificates tree.

Selecting a File that Contains the Trusted Certificate

To select the file:

1. Choose `Operations > Import Trusted Certificate` from the menu bar. The Import Trusted Certificate dialog panel appears.
2. Enter the path or folder name of the trusted certificate location.
3. Select the name of the trusted certificate file (for example, `cert.txt`).
4. Choose `OK`. A message at the bottom of the window informs you that the trusted certificate was successfully imported into the wallet.
5. Choose `OK` to exit the dialog panel; you are returned to the Oracle Wallet Manager main panel, and the trusted certificate appears at the bottom of the Trusted Certificates tree.

Removing a Trusted Certificate

To remove a trusted certificate from a wallet:

1. Select the trusted certificate listed in the Trusted Certificates tree.
2. Choose `Operations > Remove Trusted Certificate` from the menu bar.

A dialog panel warns you that your user certificate will no longer be verifiable by its recipients if you remove the trusted certificate that was used to sign it.

3. Choose `Yes`; the selected trusted certificate is removed from the Trusted Certificates tree.

Note: A certificate that is signed by a trusted certificate is no longer verifiable when you remove it from your wallet.

Also, you cannot remove a trusted certificate if it has been used to sign a user certificate that is still present in the wallet. To remove such a trusted certificate, you must first remove the certificates that it has signed.

Exporting a Trusted Certificate

To export a trusted certificate to another file system location:

1. Select `Operations > Export Trusted Certificate`; the Export Trusted Certificate dialog box appears.
2. Select a file system directory to save your trusted certificate, or choose `Browse` to display the directory structure.
3. Enter a file name to save your trusted certificate.
4. Choose `OK`; you are returned to the Oracle Wallet Manager main window.

Exporting All Trusted Certificates

To export all of your trusted certificates to another file system location:

1. Choose `Operations > Export All Trusted Certificates`. The Export Trusted Certificate dialog box appears.
2. Select the file system directory to save your trusted certificates, or choose `Browse` to display the directory structure.
3. Enter a file name to save your trusted certificates.
4. Choose `OK`; you are returned to the Oracle Wallet Manager main window.

Exporting a Wallet

You can export a wallet to text-based PKI formats. Individual components are formatted according to the following standards (Table D–2):

Table D–2 PKI Wallet Encoding Standards

Component	Encoding Standard
Certificate chains	X509v3
Trusted certificates	X509v3
Private keys	PKCS5

E

Upgrading Oracle Internet Directory

This chapter tells you how to upgrade from Oracle Internet Directory release 2.1.1.x or release 3.0.1.x to Oracle Internet Directory Release 9.0.2.1.0.

This appendix contains these topics:

Upgrading in a Single Node Environment

To upgrade on a single node, follow the instructions in the installation documentation for your operating system. See the Oracle Internet Directory section in Chapter 7 of the Oracle9*i* Application Server manual.

Upgrading in a Multi-Node Environment

Upgrading a multi-node Oracle Internet Directory system to Release 9.0.2 requires special attention.

> **See Also:** The section in *Oracle9i Application Server Administrator's Guide* about Oracle Internet Directory for more information

This section discusses LDIF-based upgrading for a multi-node Oracle Internet Directory system, which is normally not needed. Use this method when you cannot successfully run the database-based upgrade process.

LDIF-Based Upgrading

Oracle Corporation recommends that you use the LDIF-based backup procedure to backup your existing release of Oracle Internet Directory. This is explained in this section.

The LDIF-based upgrade process requires the following procedures on a node being upgraded:

Task 1: Backup the Older Version of Oracle Internet Directory

Be sure that the directory server is not running, then run the script backup_oid.sh located in the $*ORACLE_HOME*/ldap/install directory on the CD.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

The syntax to run backup_oid.sh is:

backup_oid.sh -connect *net_service_name* -pass *password_for_DB_account_'ods'*

The backup_oid.sh script does the following:

- Exports Oracle Internet Directory schema. As it does this, it generates .dmp files—for example, attr_store.dmp—in $*ORACLE_HOME*/ldap/load directory
- Backs up the Oracle Internet Directory subtree by using the ldifwrite utility. As it does this, it generates the file OID_userdata.ldif in $*ORACLE_HOME*/ldap/load. The subtree under cn=OracleSchemaVersion (if it exists) is also backed up as orcl_schemaver.ldif in the $*ORACLE_HOME*/ldap/load directory.

If you plan to install Oracle Internet Directory Release 9.0.2 in the same *ORACLE_HOME*, then save these generated files in some other location.

Task 2: Perform a Fresh Installation of Oracle Internet Directory Release 3.0.1

See Also: Installation documentation for your operating system

Task 3: Restore the User-Defined Schema and Data from the Previous Version of Oracle Internet Directory

To do this:

1. Make sure that the directory server is not running.
2. Copy the following files to $*ORACLE_HOME*/ldap/load:
   - Backed up Oracle Internet Directory schema dump files—that is, files with the extension .dmp
   - The file OID_userdata.ldif
3. Run the script restore_oid.sh located in $*ORACLE_HOME*/ldap/install.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit:http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

The syntax for restore_oid.sh is:

```
restore_oid.sh -connect net_service_name -pass password_for_DB_account_'ods'
```

The restore_oid.sh script does the following:

- Imports the Oracle Internet Directory schema from the dump files
- Inserts the schema differences between the previous release and Release 9.0.2
- Bulkloads the data from the LDIF file with the `-restore` option

Task 4: Start Oracle Internet Directory Processes

Start OID Monitor and the directory server.

See Also:

- "Starting the OID Monitor" on page 3-2
- "Starting an Oracle Directory Server Instance" on page 3-4

Upgrading a Standalone Oracle Internet Directory Node

In certain circumstances, it can be awkward to attempt an upgrade for a standalone Oracle Internet Directory node:

- When there is insufficient disk space to install and upgrade OID 9.0.2.1.0 on the machine where the prior version of Oracle Internet Directory resides, or
- When the amount of data in the prior Oracle Internet Directory version is huge, and it is more comfortable to do data export/import than to do database migration.

Note: If the Oracle Internet Directory node is taking part in replication or is configured for replication, then this procedure should not be used to upgrade the node.

To upgrade the directory in these circumstances, follow the steps in the sections listed here:

- Task 1: Stop Oracle Directory Server on the Old Version Node
- Task 2: Backup the Sponsor Node by Using Export Utility
- Task 3: Load Data into the New Node by Using the Import Utility
- Task 4: Perform Oracle Internet Directory Schema Upgrade

Task 1: Stop Oracle Directory Server on the Old Version Node

To stop the Oracle directory server, run the following command from $*ORACLE_HOME*/bin/ on the node:

```
oidctl connect=<db_connect_string> server=oidldapd instance=1 stop
```

Task 2: Backup the Sponsor Node by Using Export Utility

1. Create a new file, `oidexp.dat`, containing the following:

   ```
   FILE=oid.data
   OWNER=ods, odscommon
   RANTS=y
   ROWS=y
   ```

2. Run the following command (from $ORACLE_HOME/bin/) against the identified sponsor node:

   ```
   exp system/manager PARFILE=oidexp.dat
   ```

   Note: OID schema and data will be backed up in oid.data file. Move this file to the new node before performing the next task.

Task 3: Load Data into the New Node by Using the Import Utility

1. Run the following SQL scripts:

   ```
   cd $ORACLE_HOME/ldap/admin/
   sqlplus system/manager @ldapdrop.sql
   sqlplus system/manager @ldapxact.sql
   sqlplus system/manager @ldapxsec.sql
   ```

2. Create a new file, `oidimp1.dat`, containing the following:

```
FILE=oid.data
FROMUSER=ods
TOUSER=ods
```

3. Run the following commands against the new node:

```
imp system/manager PARFILE=oidimp1.dat
```

> **Note:** Make sure the backup `oid.data` file is present in the current directory:

4. Create a new file, oidimp2.dat, containing the following:

```
FILE=oid.data
FROMUSER=odscommon
TOUSER=odscommon
```

5. Run the following commands against the new node:

```
imp system/manager PARFILE=oidimp2.dat
```

> **Note:** Make sure the backup `oid.data` file is present in the current directory:

Task 4: Perform Oracle Internet Directory Schema Upgrade

1. Launch the OID Configuration Assistant by running $*ORACLE_HOME*/bin/oidca
2. Click Next at the Welcome Screen.
3. Select the option 'Upgrade an existing OID' and click Next.
4. The Database Migration Screen comes up. Here you are required to provide the information about the database to which Oracle Internet Directory data has been imported. (New version OID). Enter following information
   a. Database SID
   b. Passwords for the database users, 'SYSTEM' and 'ODS' respectively
   c. Oracle Home

d. Location of the INIT.ORA file.

5. Listener Port for the OID database

f. Connect String for the OID database

5. Click Next. (When this operation completes, the Oracle Internet Directory base schema has been upgraded to Oracle Internet Directory version 9.0.2.1.0.)
6. In the next screen, provide the following information about the Oracle directory server:
   a. Non-SSL port on which the directory server needs to be started. The default value specified is 389.
   b. SSL port on which the directory server needs to be started. The default value specified is 636.
   c. The super-user distinguished name.
   d. The corresponding super-user password.
7. Click Next. In the next step, Oracle-context-related information and the Oracle Directory Integration Platform-related information will be upgraded.
8. The Upgrading Subscriber screen appears. Here, you need to enter the distinguished name (DN) that identifies the root of your organization—for example, `o=acme, dc=com`. This domain is then upgraded to become the default subscriber.
9. Click Next. The User Data Migration screen appears. This step might take a long time if you have a large directory. If you have a large directory—that is, more than 10,000 users—then Oracle Corporation recommends that you postpone your data migration and do it as a post-install step.
10. However, if you want to do the user data migration as a part of this OID Configuration Assistant operation, select Yes and click Next. This completes the user data migration.

At the end of the upgrade, the directory server is running, listening to the specified Non-SSL and SSL ports.

F

Migrating Data from Other LDAP-Compliant Directories

This appendix tells how to migrate data from both LDAP Version 3-compatible directories and application-specific directories into Oracle Internet Directory.

This appendix contains these topics:

About the Data Migration Process

You can import data from a third-party LDAP-compliant directory into Oracle Internet Directory by saving the data in an LDIF file. LDIF is the IETF-sanctioned ASCII interchange format for representing LDAP-compliant directory data as a file. All LDAP-compliant directories should be able to export their contents into one or more LDIF files representing the DIT at the time of export.

Be aware that certain proprietary attributes or metadata may be included in a given product's LDIF output. You must remove this extraneous data from the LDIF file before you import the file into Oracle Internet Directory. In such cases, you need to perform some additional steps before importing the LDIF files into Oracle Internet Directory. The next section explains these steps.

> **See Also:** The LDIF technical specification available for download at:`http://www.ietf.org/rfc/rfc2849.txt`

Tasks For Migrating Data from LDAP-Compliant Directories

To migrate data from LDAP-compliant directories, you perform these tasks:

- Export Data from the Non-Oracle Internet Directory Server into LDIF File Format
- Analyze the LDIF User Data for Any Required Schema Additions Referenced in the LDIF Data
- Extend the Schema in Oracle Internet Directory
- Remove Any Proprietary Directory Data from the LDIF File
- Remove Operational Attributes from the LDIF File
- Remove Incompatible userPassword Attribute Values from the LDIF File
- Run the bulkload.sh -check Mode and Determine Any Remaining Schema Violations or Duplication Errors

Task 1: Export Data from the Non-Oracle Internet Directory Server into LDIF File Format

See the vendor-supplied documentation for instructions. If flags or options exist for exporting data from the foreign directory, be sure to select the method that:

- Produces LDIF output with the least amount of proprietary information included

- Provides maximum conformance to the IETF Request for Comments 2849 mentioned in About the Data Migration Process on page F-2

Task 2: Analyze the LDIF User Data for Any Required Schema Additions Referenced in the LDIF Data

Any attributes not found in the Oracle Internet Directory base schema require extension of the Oracle Internet Directory base schema prior to the importation of the LDIF file. Some directories may support the use of configuration files for defining extensions to their base schema (Oracle Internet Directory does not). If you have a configuration file you can use it as a guideline for extending the base schema in Oracle Internet Directory in "Task 3: Extend the Schema in Oracle Internet Directory".

Task 3: Extend the Schema in Oracle Internet Directory

See Chapter 6, "Directory Schema Administration" for tips on how to extend the directory schema in Oracle Internet Directory. You can do this by using either Oracle Directory Manager or the SchemaSynch tool as explained in "SchemaSync Syntax" on page A-57.

Task 4: Remove Any Proprietary Directory Data from the LDIF File

Certain elements of the LDAP v3 standard have not yet been formalized, such as **ACI** attributes. As a result, various directory vendors implement ACI policy objects in ways that do not translate well across vendor installations.

After the basic entry data has been imported from the cleaned up LDIF file to Oracle Internet Directory, you must explicitly reapply security policies in the Oracle Internet Directory environment. You can do this by using either Oracle Directory Manager, or command-line tools and LDIF files containing the desired **ACP** information.

There may be other proprietary metadata unrelated to access control. You should remove this as well. Understanding the various IETF RFCs can help you determine which directory metadata is proprietary to a given vendor and which complies with the LDAP standards, and is thus portable by way of an LDIF file.

Task 5: Remove Operational Attributes from the LDIF File

Four of the standard LDAP v3 operational attributes, namely, `creatorsName`, `createTimestamp`, `modifiersName`, and `modifyTimestamp` are automatically

generated by Oracle Internet Directory whenever entries are created or imported. It is not possible to instantiate these values from existing directory data, for example by using LDIF file importation. Therefore you should remove these attributes from the file before attempting to import.

Task 6: Remove Incompatible userPassword Attribute Values from the LDIF File

Oracle Internet Directory Release 9.0.2 supports the following `userPassword` attribute hash algorithms:

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

The `userPassword` attribute hash values used by some vendor products are not compatible with Oracle Internet Directory. As a result, you must remove all lines corresponding to the `userPassword` attribute and value from the LDIF data file unless they are represented in plain text or contain no value. After importation of the LDIF data, you must re-enter manually or upload hashed `userPassword` information separately into the directory.

Task 7: Run the bulkload.sh -check Mode and Determine Any Remaining Schema Violations or Duplication Errors

Before generating and loading an LDIF file, always perform a check on it by using the bulkload utility check mode. The bulkload output reports any inconsistencies in the data.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.0. Visit: http://sources.redhat.com/cygwin/
- MKS Toolkit 5.1 or 6.0. Visit: http://www.datafocus.com/products/

See Also: bulkload Syntax on page A-35 for instructions on how to use the bulkload check mode

G

The LDAP Filter Definition

The paper contained in this appendix is copied with permission from The Internet Engineering Task Force. The URL for this document is:
`http://www.ietf.org/rfc/rfc2254.txt`

The contents of this paper may have been superseded by later papers or other information. Check the above Web site and related sites for additional or supplementary information.

NOTE: ORACLE DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Network Working Group T. Howes
Request for Comments: 2254 Netscape Communications Corp.
Category: Standards Track December 1997

The String Representation of LDAP Search Filters

1. Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

IESG Note

This document describes a directory access protocol that provides both read and update access. Update access requires secure authentication, but this document does not mandate implementation of any satisfactory authentication mechanisms.

In accordance with RFC 2026, section 4.4.1, this specification is being approved by IESG as a Proposed Standard despite this limitation, for the following reasons:

a. to encourage implementation and interoperability testing of these protocols (with or without update access) before they are deployed, and

b. to encourage deployment and use of these protocols in read-only applications. (e.g. applications where LDAPv3 is used as a query language for directories which are updated by some secure mechanism other than LDAP), and

c. to avoid delaying the advancement and deployment of other Internet standards-track protocols which require the ability to query, but not update, LDAPv3 directory servers.

Readers are hereby warned that until mandatory authentication mechanisms are standardized, clients and servers written according to this specification which make use of update functionality are UNLIKELY TO INTEROPERATE, or MAY INTEROPERATE ONLY IF AUTHENTICATION IS REDUCED TO AN UNACCEPTABLY WEAK LEVEL.

Implementors are hereby discouraged from deploying LDAPv3 clients or servers which implement the update functionality, until a Proposed Standard for mandatory authentication in LDAPv3 has been approved and published as an RFC.

2. Abstract

The Lightweight Directory Access Protocol (LDAP) [1] defines a network representation of a search filter transmitted to an LDAP server. Some applications may find it useful to have a common way of representing these search filters in a human-readable form. This document defines a human-readable string format for representing LDAP search filters.

This document replaces RFC 1960, extending the string LDAP filter definition to include support for LDAP version 3 extended match filters, and including support for representing the full range of possible LDAP search filters.

3. LDAP Search Filter Definition

An LDAPv3 search filter is defined in Section 4.5.1 of [1] as follows:

```
Filter ::= CHOICE {
        and             [0] SET OF Filter,
        or              [1] SET OF Filter,
        not             [2] Filter,
        equalityMatch   [3] AttributeValueAssertion,
        substrings      [4] SubstringFilter,
        greaterOrEqual  [5] AttributeValueAssertion,
        lessOrEqual     [6] AttributeValueAssertion,
        present         [7] AttributeDescription,
        approxMatch     [8] AttributeValueAssertion,
        extensibleMatch [9] MatchingRuleAssertion
}
SubstringFilter ::= SEQUENCE {
        type    AttributeDescription,
        SEQUENCE OF CHOICE {
```

```
            initial      [0] LDAPString,
            any          [1] LDAPString,
            final        [2] LDAPString
        }
    }
    AttributeValueAssertion ::= SEQUENCE {
        attributeDesc   AttributeDescription,
        attributeValue  AttributeValue
    }
    MatchingRuleAssertion ::= SEQUENCE {
        matchingRule    [1] MatchingRuleID OPTIONAL,
        type            [2] AttributeDescription OPTIONAL,
        matchValue      [3] AssertionValue,
        dnAttributes    [4] BOOLEAN DEFAULT FALSE
    }
    AttributeDescription ::= LDAPString
    AttributeValue ::= OCTET STRING
    MatchingRuleID ::= LDAPString
    AssertionValue ::= OCTET STRING
    LDAPString ::= OCTET STRING
```

where the LDAPString above is limited to the UTF-8 encoding of the ISO 10646 character set [4]. The AttributeDescription is a string representation of the attribute description and is defined in [1].

The AttributeValue and AssertionValue OCTET STRING have the form defined in [2]. The Filter is encoded for transmission over a network using the Basic Encoding Rules defined in [3], with simplifications described in [1].

4. String Search Filter Definition

The string representation of an LDAP search filter is defined by the following grammar, following the ABNF notation defined in [5]. The filter format uses a prefix notation.

```
filter     = "(" filtercomp ")"
filtercomp = and / or / not / item
and        = "&" filterlist
or         = "|" filterlist
not        = "!" filter
filterlist = 1*filter
item       = simple / present / substring / extensible
simple     = attr filtertype value
filtertype = equal / approx / greater / less
equal      = "="
approx     = "~="
greater    = ">="
less       = "<="
extensible = attr [":dn"] [":" matchingrule] ":=" value
           / [":dn"] ":" matchingrule ":=" value
present    = attr "=*"
substring  = attr "=" [initial] any [final]
initial    = value
any        = "*" *(value "*")
final      = value
attr       = AttributeDescription from Section 4.1.5 of [1]
matchingrule = MatchingRuleId from Section 4.1.9 of [1]
value      = AttributeValue from Section 4.1.6 of [1]
```

The attr, matchingrule, and value constructs are as described in the corresponding section of [1] given above.

If a value should contain any of the following characters

Character	ASCII value

*	0x2a
(	0x28
)	0x29
\	0x5c
NUL	0x00

the character must be encoded as the backslash '\' character (ASCII 0x5c) followed by the two hexadecimal digits representing the ASCII value of the encoded character. The case of the two hexadecimal digits is not significant.

This simple escaping mechanism eliminates filter-parsing ambiguities and allows any filter that can be represented in LDAP to be represented as a NUL-terminated string. Other characters besides the ones listed above may be escaped using this mechanism, for example, non-printing characters.

For example, the filter checking whether the "cn" attribute contained a value with the character "*" anywhere in it would be represented as

"(cn=*\2a*)".

Note that although both the substring and present productions in the grammar above can produce the "attr=*" construct, this construct is used only to denote a presence filter.

5. Examples

This section gives a few examples of search filters written using this notation.

(cn=Babs Jensen)

(!(cn=Tim Howes))

(&(objectClass=Person)(|(sn=Jensen)(cn=Babs J*)))

(o=univ*of*mich*)

The following examples illustrate the use of extensible matching.

(cn:1.2.3.4.5:=Fred Flintstone)

(sn:dn:2.4.6.8.10:=Barney Rubble)

(o:dn:=Ace Industry)

(:dn:2.4.6.8.10:=Dino)

The second example illustrates the use of the ":dn" notation to indicate that matching rule "2.4.6.8.10" should be used when making comparisons, and that the

attributes of an entry's distinguished name should be considered part of the entry when evaluating the match.

The third example denotes an equality match, except that DN components should be considered part of the entry when doing the match.

The fourth example is a filter that should be applied to any attribute supporting the matching rule given (since the attr has been left off). Attributes supporting the matching rule contained in the DN should also be considered.

The following examples illustrate the use of the escaping mechanism.

(o=Parens R Us \28for all your parenthetical needs\29)

(cn=*\2A*)

(filename=C:\5cMyFile)

(bin=\00\00\00\04)

(sn=Lu\c4\8di\c4\87)

The first example shows the use of the escaping mechanism to represent parenthesis characters. The second shows how to represent a "*" in a value, preventing it from being interpreted as a substring indicator. The third illustrates the escaping of the backslash character.

The fourth example shows a filter searching for the four-byte value 0x00000004, illustrating the use of the escaping mechanism to represent arbitrary data, including NUL characters.

The final example illustrates the use of the escaping mechanism to represent various non-ASCII UTF-8 characters.

6. Security Considerations

This memo describes a string representation of LDAP search filters. While the representation itself has no known security implications, LDAP search filters do. They are interpreted by LDAP servers to select entries from which data is retrieved. LDAP servers should take care to protect the data they maintain from unauthorized access.

7. References

[1] Wahl, M., Howes, T., and S. Kille, "Lightweight Directory Access

Protocol (v3)", RFC 2251, December 1997.

[2] Wahl, M., Coulbeck, A., Howes, T., and S. Kille, "Lightweight

Directory Access Protocol (v3): Attribute Syntax Definitions", RFC 2252, December 1997.

[3] Specification of ASN.1 encoding rules: Basic, Canonical, and Distinguished Encoding Rules, ITU-T Recommendation X.690, 1994.

[4] Yergeau, F., "UTF-8, a transformation format of Unicode and ISO 10646", RFC 2044, October 1996.

[5] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, August 1982.

8. Author's Address

Tim Howes
Netscape Communications Corp.
501 E. Middlefield Road
Mountain View, CA 94043
USA
Phone: +1 415 937-3419
EMail: howes@netscape.com

9. Full Copyright Statement

BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

H

Troubleshooting

This appendix explains typical problems that you could encounter while running or installing Oracle Internet Directory. It contains these topics:

- Installation Errors
- Administration Error Messages and Causes

Installation Errors

During installation and configuration of the Oracle9*i* database server, you must select the character set UTF-8. If you select any other character set, the directory server will not function properly.

Administration Error Messages and Causes

This section contains a list of all the Oracle directory server error messages that you can encounter. Each message is followed by its most probable causes.

This section contains these topics:

Oracle Database Server Error Due to Schema Modifications

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 H-6.

00—LDAP_SUCCESS

Cause: The operation was successful.

01—LDAP_OPERATIONS_ERROR

Cause: General errors encountered by the server when processing the request.

02—LDAP_PROTOCOL_ERROR

Cause: The client request did not meet the LDAP protocol requirements, such as format or syntax. This can occur in the following situations:

- Server encounters a decoding error while parsing the incoming request
- The request is an add or modify request that specifies the addition of an attribute type to an entry but no values specified

- Error reading SSL credentials
- An unknown type of modify operation is specified (other than LDAP_MOD_ADD, LDAP_MOD_DELETE, and LDAP_MOD_REPLACE)
- Unknown search scope

03—LDAP_TIMELIMIT_EXCEEDED

Cause: Search took longer than the time limit specified. If you have not specified a time limit for the search, Oracle Internet Directory uses a default time limit of one hour.

04—LDAP_SIZELIMIT_EXCEEDED

Cause: More entries match the search query than the size limit specified. If you have not specified a size limit for the search, Oracle Internet Directory uses a default size limit.

05—LDAP_COMPARE_FALSE

Cause: Presented value is not the same as the one in the entry.

06—LDAP_COMPARE_TRUE

Cause: Presented value is same as the one in the entry.

07—LDAP_STRONG_AUTH_NOT_SUPPORTED

Cause: Bind method is not supported by the server.

08—LDAP_STRONG_AUTH_REQUIRED

Cause: Strong authentication is required. Oracle Internet Directory does not return this message at the present time.

09—LDAP_PARTIAL_RESULTS

Cause: Server returned a referral.

10—LDAP_REFERRAL

Cause: Server returned a referral.

11—LDAP_ADMINLIMIT_EXCEEDED

Cause: Oracle Internet Directory does not return this message at the present time.

12—LDAP_UNAVAILABLE_CRITICALEXTENSION

Cause: Specified request is not supported

16—LDAP_NO_SUCH_ATTRIBUTE

Cause: Attribute does not exist in the entry specified in the request.

17—LDAP_UNDEFINED_TYPE

Cause: Specified attribute type is undefined in the schema.

18—LDAP_INAPPROPRIATE_MATCHING

Cause: Specified matching rule is inappropriate for the attribute type. Oracle Internet Directory does not return this message at the present time.

19—LDAP_CONSTRAINT_VIOLATION

Cause: The value in the request violated certain constraints.

20—LDAP_TYPE_OR_VALUE_EXISTS

Cause: Duplicate values specified for the attribute.

21—LDAP_INVALID_SYNTAX

Cause: Specified *attribute* syntax is invalid. In a search, the *filter* syntax is invalid.

32—LDAP_NO_SUCH_OBJECT

Cause: The base specified for the operation does not exist.

33—LDAP_ALIAS_PROBLEM

Cause: Oracle Internet Directory does not return this message at the present time.

34—LDAP_INVALID_DN_SYNTAX

Cause: Error in the DN syntax.

35—LDAP_IS_LEAF

Cause: The entry is a leaf (terminal entry). Oracle Internet Directory does not return this message at the present time.

36—LDAP_ALIAS_DEREF_PROBLEM

Cause: Oracle Internet Directory does not return this message at the present time.

48—LDAP_INAPPROPRIATE_AUTH

Cause: Oracle Internet Directory does not return this message at the present time.

49—LDAP_INVALID_CREDENTIALS

Cause: Bind failed because the credentials are not correct.

50—LDAP_INSUFFICIENT_ACCESS

Cause: The client does not have access to perform this operation.

51—LDAP_BUSY

Cause: Server cannot accept any more client connections. Oracle Internet Directory does not return this message at the present time.

52—LDAP_UNAVAILABLE

Cause: Cannot contact the server at all. Oracle Internet Directory does not return this message at the present time.

53—LDAP_UNWILLING_TO_PERFORM

Cause: General error, or server is in read-only mode.

54—LDAP_LOOP_DETECT

Cause: Oracle Internet Directory does not return this message at the present time.

64—LDAP_NAMING_VIOLATION

Cause: Oracle Internet Directory does not return this message at the present time.

65—LDAP_OBJECT_CLASS_VIOLATION

Cause: A change to the entry violates the objectclass definition.

66— LDAP_NOT_ALLOWED_ON_NONLEAF

Cause: The entry to be deleted has children.

67—LDAP_NOT_ALLOWED_ON_RDN

Cause: Cannot perform the operation on RDN attributes—for example, you cannot delete the RDN attribute of the entry.

68—LDAP_ALREADY_EXISTS

Cause: Duplicate ADD condition.

69—LDAP_NO_OBJECT_CLASS_MODS

Cause: Oracle Internet Directory does not return this message at the present time.

70—LDAP_RESULTS_TOO_LARGE

Cause: Oracle Internet Directory does not return this message at the present time.

80—LDAP_OTHER

Cause: Oracle Internet Directory does not return this message at the present time.

81—LDAP_SERVER_DOWN

Cause: Can't contact LDAP server. This message is returned from the SDK.

82—LDAP_LOCAL_ERROR

Cause: The client encountered an internal error. This message is returned from the client SDK.

83—LDAP_ENCODING_ERROR

Cause: The client encountered an error in encoding the request. This message is returned from the SDK.

84—LDAP_DECODING_ERROR

Cause: The client encountered an error in decoding the request. This message is returned from the SDK.

85—LDAP_TIMEOUT

Cause: Client encountered the time-out specified for the operation. This message is returned from the SDK.

86—LDAP_AUTH_UNKNOWN

Cause: Authentication method is unknown to the client SDK.

87—LDAP_FILTER_ERROR

Cause: Bad search filter

88—LDAP_USER_CANCELLED

Cause: User cancelled operation

89—LDAP_PARAM_ERROR

Cause: Bad parameter to an LDAP routine

90—LDAP_NO_MEMORY

Cause: Out of memory

Additional Error Messages

These messages do not display error codes.

The Oracle Internet Directory application replaces the *parameter* tag seen in some of the messages below with the appropriate run-time value.

%s attribute not found.

Cause: The particular attribute type is not defined in the schema.

<parameter> not found for attribute <parameter>.

Cause: Value not found in the attribute. (ldapmodify)

Admin domain does not contain schema information for objectclass <parameter>.

Cause: The object class specified in the request is not present in the schema.

Attempted to add a Class with oid <parameter> taken by other class.

Cause: Duplicate object identifier specified. (schema modification)

Attribute <parameter> already in use.

Cause: Duplicate attribute name. (schema modification)

Attribute <parameter> has syntax error.

Cause: Syntax error in the attribute name definition. (schema modification)

Attribute <parameter> is not supported in the schema.

Cause: Attribute not defined. (all operations)

Attribute <parameter> is single valued.

Cause: Attribute is single-valued. (ldapadd & ldapmodify)

Attribute <parameter> not present in the entry.

Cause: This attribute does not exist in the entry. (ldapmodify)

Bad attribute definition.

Cause: Syntax error in attribute definition. (schema modification)

Currently Not Supported

Cause: The version of LDAP request is not supported by this server.

Entry to be deleted not found.

Cause: DN specified in the delete operation not found.

Entry to be modified not found

Cause: The entry specified in the request is not found.

Error encountered while adding <parameter> to the entry

Cause: Returned when modify add operation is invoked. A possible cause is that the system resource is unavailable.

Error encountered while encrypting an attribute value.

Cause: Error in encrypting user password. (all operations)

Error in DN Normalization.

Cause: DN specified is invalid. Syntax error encountered in parsing the DN. (all operations)

Error in hashing <parameter> attribute.

Cause: Error in creating hash entry for the attribute. (schema modification)

Error in hashing <parameter> objectclass.

Cause: Error in creating hash entry for the objectclass. (schema modification)

Error in Schema hash creation.

Cause: Error while creating hash table for schema. (schema modification)

Error replacing <parameter>.

Cause: Error in replacing this attribute. (ldapmodify)

Error while normalizing value for attribute <parameter>.

Cause: Error in normalizing value for the attribute. (all operations)

Failed to find <parameter> in mandatory or optional attribute list.

Cause: Attribute specified does not exist in either the mandatory or optional attribute list as required by the object class(es).

Function Not Implemented

Cause: The feature / request is currently not supported.

INVALID ACI is <parameter>

Cause: The particular ACI you specified in a request is invalid.

Mandatory attribute <parameter> is not defined in Admin Domain <parameter>.

Cause: MUST refers to attribute not defined. (schema modification)

Mandatory Attribute missing.

Cause: The mandatory attribute for the particular entry is missing, as required by the particular object class.

Matching rule, <parameter>, not defined.

Cause: Matching rule not defined in the server. (schema modification)

MaxConn Reached

Cause: The maximum number of concurrent connections to the LDAP server has been reached.

Modifying the Naming attribute for the entry without modifying the DN.

Cause: Cannot modify the naming attributes using ldap_modify. A naming attribute, such as *cn* is an element in the DN.

New Parent not found.

Cause: New parent specified in modifydn operation does not exist.(ldapmodifydn)

Object already exists.

Cause: Duplicate entry. (ldapadd and ldapmodifydn)

Object ID <parameter> already in use.

Cause: Duplicate object identifier specified. (schema modification)

Objectclass <parameter> already in use. m

Cause: Duplicate Objectclass name. (schema modification)

Objectclass attribute missing.

Cause: The objectclass attribute is missing for this particular entry.

OID <parameter> has syntax error.

Cause: syntax error in the object identifier definition. (schema modification)

One of the attributes in the entry has duplicate value

Cause: You entered two values for the same attribute in the entry you are creating.

Operation not allowed on the <parameter>.

Cause: Operation not allowed on this entry. (modify, add, and delete)

Operation not allowed on the DSE Entry.

Cause: Can't do this operation on DSE entry. (delete)

Optional attribute <parameter> is not defined in Admin Domain <parameter>.

Cause: MAY refers to attribute not defined. (schema modification)

Parent entry not found in the directory.

Cause: Parent entry does not exist. (ldapadd and perhaps ldapmodifydn)

Super object <parameter> is not defined in Admin Domain <parameter>.

Cause: SUP types refer to non-existing class. (schema modification)

Super type undefined.

Cause: SUP type does not exist. (schema modification)

Super user addition not permitted.

Cause: Cannot create super user entry. (ldapadd)

Syntax, <parameter>, not defined.

Cause: Syntax not defined in the server. (schema modification)

The attribute or the value specified in the RDN does not exist in the entry.

Cause: AVA specified as the RDN does not exist in the entry. (ldapadd)

Unknown search scope

Cause: The search scope specified in the LDAP request is not recognized.

Version Not Supported

Cause: The version of the LDAP request is not supported by this server.

Password Policy Violation Error Messages

Table 36–6 contains the error messages that are sent to the client as a result of password policy violations. The error codes are not standard LDAP error codes. They are messages sent as a part of additional information in the LDAP result.

Table 36–6 Password Policy Violation Error Messages

Error Number	Exception	Comment or Resolution
9000	GSL_PWDEXPIRED_EXCP	Your Password has expired. Please contact the Administrator to change your password.
9001	GSL_ACCOUNTLOCKED_EXCP	Your account is locked. Please contact the Administrator.
9002	GSL_EXPIREWARNING_EXCP	Your Password will expire in `pwdexpirewarning` seconds. Please change your password now.
9003	GSL_PWDMINLENGTH_EXCP	Your Password must be at least `pwdminlength` characters long.
9004	GSL_PWDNUMERIC_EXCP	Your Password must contain at least `orclpwdalphanumeric` numeric characters.
9005	GSL_PWDNULL_EXCP	Your Password cannot be a Null Password.
9006	GSL_PWDINHISTORY_EXCP	Your New Password cannot be the same as your Old Password.
9007	GSL_PWDILLEGALVALUE_EXCP	Your Password cannot be the same as your `orclpwdillegalvalues`.
9008	GSL_GRACELOGIN_EXCP	Your Password has expired. You have `pwdgraceloginlimit` Grace logins left.
9050	GSL_ACCTDISABLED_EXCP	Your Account has been disabled. Please contact the administrator.

I

Migrating User Data from Application-Specific Repositories

This chapter contains these topics:

About Migrating from Application-Specific Repositories

Migrating user data from an application-specific repository requires:

- Collecting the user data from the application-specific repository and formatting it in a way that the directory can read it
- Making that data available to the directory administrator who must then:
  - Specify where to place it in the directory
  - Import it into the directory

To enable this migration to happen, the Oracle Directory Provisioning Integration Service relies on the application-specific repository exporting its data to an intermediate template file. This is not a pure LDIF file. Rather, records in this template file are in LDIF, but with substitution variables that the application itself leaves undefined—for you, the directory administrator, to define later in the process. These variables have to do with, for example, the location in the directory where the information is finally to reside.

To convert the user data from this intermediate template file into proper LDIF, you use the OID Migration Tool. Once the data is converted to LDIF, you can load it into the directory.

To summarize: Migrating data from application-specific repositories involves these general steps:

1. (LDIF) template file
2. You, the directory administrator, using the OID Migration Tool to read these partial LDIF entries and convert them to actual LDIF entries based on the deployment choices
3. You, the directory administrator, loading the data, now in LDIF, into Oracle Internet Directory.
4. The application completing the migration process according to its own specifications.

Tasks For Migrating Data from Application-Specific Repositories

You can run the OID Migration Tool in one of two modes:

- Simple mode, in which you specify all values of the substitution variables
- Look-up mode, in which the OID Migration Tool determines values for certain substitution variables by searching the directory

To migrate data from application-specific repositories, you create an intermediate template file, then run the OID Migration Tool.

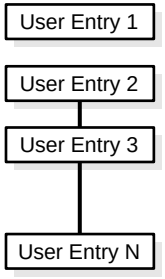
Task 1: Create an Intermediate Template File

Applications generating data in national languages must store that data in AL32UTF8 in the intermediate template file as specified in the IETF RFC 2849, "The LDAP Data Interchange Format (LDIF) - Technical Specification" available at http://www.ietf.org/rfc/rfc2849.txt.

When generating the intermediate template file, migrating applications must list all user records sequentially with a record separator as defined in RFC 2849. The OID User Migration Tool assigns all of these users to the default subscriber, which corresponds to the enterprise itself.

Figure I–1 shows the overall structure of the intermediate template file containing user entries.

Figure I–1 Structure of the Intermediate User File



The intermediate template file uses the following format to generate a valid user entry. All of the strings in **`bold text`** are supplied from the application-specific repository.

```
dn: cn=UserID, %s_UserContainerDN%
sn: Last_Name
orclGlobalID: GUID_for_User
%s_UserNicknameAttribute%: UserID
objectClass: inetOrgPerson
objectClass: orclUserV2
```

In this template, the strings ***%s_UserContainerDN%*** and ***%s_UserNicknameAttribute%*** are substitution variables for which the OID

Migration Tool provides values. The OID Migration Tool determines these values according to deployment-specific considerations. Either the application passes the arguments to the OID Migration Tool, or the tool retrieves them from the directory.

Example: User Entries in an Intermediate Template File

The following intermediate template file includes user entries generated by the application-specific migration logic. In this example, all of the data listed in **bold text** is from the application-specific user repository.

```
dn: cn=jdoe, %s_UserContainerDN%
sn: Doe
%s_UserNicknameAttribute%: jdoe
objectClass: inetOrgPerson
objectClass: orclUserV2
title: Member of Technical Staff
homePhone: 415-584-5670
homePostalAddress: 234 Lez Drive$ Redwood City$ CA$ 94402


dn: cn=jsmith, %s_UserContainerDN%
sn: Smith
%s_UserNicknameAttribute%: jsmith
objectClass: inetOrgPerson
objectClass: orclUserV2
title: Member of Technical Staff
homePhone: 650-584-5670
homePostalAddress: 232 Gonzalez Drive$ San Francisco$ CA$ 94404


dn: cn=lrider, %s_UserContainerDN%
sn: Rider
%s_UserNicknameAttribute%: lrider
objectClass: inetOrgPerson
objectClass: orclUserV2
title: Senior Member of Technical Staff
homePhone: 650-584-5670
```

Once all of the user data is converted to the intermediate file format, the OID Migration Tool further converts it into a proper LDIF file that can be loaded into Oracle Internet Directory.

You can find examples of intermediate template files in $*SRCHOME*/ldap/schema/oid.

Attributes in User Entries

Each user entry has mandatory and optional attributes.

Table I–1 lists and describes the mandatory attributes in a user entry.

Table I–1 Mandatory Attributes in a User Entry

Attribute	Description
dn	Distinguished name of the user entry with appropriate substitution variables. The relative distinguished name of the entry MUST be cn.
sn	Surname—that is, the last name—of the user
objectclass	Object classes the entry should minimally belong to: inetOrgPerson and orclUserV2

The following are optional attributes from the inetOrgPerson object class:

orclGuid
userPassword
telephoneNumber
seeAlso
description
title
x121Address
registeredAddress
destinationIndicator
preferredDeliveryMethod
telexNumber
teletexTerminalIdentifier
internationaliSDNNumber
facsimileTelephoneNumber
street
postOfficeBox
postalCode
postalAddress
physicalDeliveryOfficeNameou
st
l
audio
businessCategory
carLicense
departmentNumber
displayName
employeeNumber
employeeType
givenName
homePhone
homePostalAddress
initials
jpegPhoto
labeledURI
mail
manager
mobile
pager
photo
preferredLanguage
roomNumber
secretary
uid
userCertificate
x500UniqueIdentifier
userSMIMECertificate
userPKCS12

See Also: IETF Request for Comments 2798: "Definition of the `inetOrgPerson` LDAP Object Class," available at `http://www.ietf.org/rfc/rfc2798.txt?number=2798`, for a description of each attribute in this object class

The following are optional attributes from the `orclUserV2` object class:

Table I–2 Attributes in the orclUserV2 Object Class

Attribute	Description
`OrclPassword`	An Oracle-specific password identifier for custom authentication schemes like O3Logon for the database server
`OrclHireDate`	Specifies the date on which an employee starts working for a company or subscriber
`OrclDefaultProfileGroup`	Holds the name (DN) of the group to designate a default group for a user such that a default profile can be built for the user based on this attribute value.
`OrclPasswordHint`	Specifies the question set by a user for administering password on behalf of a user
`OrclPasswordHintAnswer`	Specifies the answer set for `orclPasswordHint`
`OrclTimeZone`	Indicates the geographical time zone of a user based on his office location.Valid values are the three letter time zone values—for example, EST, PST, GMT
`OrclIsVisisble`	Specifies whether the user entry should be displayed in people search applications
`OrclDisplayPersonalInfo`	Specifies if the user personal information should be displayed in white pages queries
`OrclWorkflowNotificationPref`	Specifies the preferred notification mechanism for Oracle Workflow.
`OrclMaidenName`	Specifies the maiden name of an individual
`OrclDateOfBirth`	Specifies the date on which an individual was born
`orclActiveStartDate`	The date on which the user can successfully begin to authenticate to the Oracle9*i*AS Single Sign-On server. Values are represented in Universal Time format.

Table I–2 Attributes in the orclUserV2 Object Class

Attribute	Description
orclEnddate	The date after which the user can no longer authenticate to the Oracle9*i*AS Single Sign-On server. Values are represented in Universal Time format.

Task 2: Run the OID Migration Tool

Once you have set up the intermediate template file, the OID Migration Tool, described in the next section, enables you to bring all pertinent data from the application-specific repository into Oracle Internet Directory. Once you have migrated the data, you can update whatever portion of it is relevant to the application by synchronizing that application with Oracle Internet Directory. You synchronize by using either the Oracle Directory Synchronization Service or the Oracle Directory Provisioning Integration Service.

The OID Migration Tool

Use the OID Migration Tool when you are migrating data from application-specific repositories into Oracle Internet Directory. The OID Migration Tool produces an LDIF file, which is suitable for loading into a directory server by using the standard command-line tools. The input to this tool is a pseudo-LDIF file containing substitution variables. The tool is called ldifmigrator and it exists in *ORACLE_HOME*/bin.

The syntax of the ldifmigrator tool is as follows:

```
$ ldifmigrator Input_file=my_users.dat" "Output_file=my_users.ldif"
        [-lookup "Host=directoryName"
        ["Port=portnumber"]
        "DN=bindDn"
        ["Password=password"]
        ["Subscriber=subscribername"]]
        {"s_SubVar1=val1" ..."s_SubVarN=valN"  }
```

Table I–3 describes the command-line parameters used by this tool in further detail:

Table I–3 ldifmigrator Parameters

Parameter	Mandatory/Optional	Description
Input_file	M	The file containing the substitution variables

Table I–3 Idifmigrator Parameters

Parameter	Mandatory/Optional	Description
Output_file	M	The Name of the file to be generated by this tool
-lookup	O	If this flag is specified, then values of certain substitution variables will be obtained from the directory server. Please see the following table for the names of the variables that are The name of the directory server is specified using host parameter. The host is mandatory when –lookup flag is specified.
Host	M (only in lookup mode)	The directory server name. This parameter is mandatory when –lookup flag is specified.
Port	O	The port on which the directory server is listening. If not specified the port 389 will be used
DN	M (only in lookup mode)	Bind DN. This is a mandatory parameter when -lookup flag is specified.
Password	O	Bind password
Subscriber	O	The subscriber whose attributes will be used as substitution variable. If not specified the default subscriber specified in the Root Oracle Context will be used
s_SubsVar1..N	O	Custom substitution variables specified by the user.

The following table describes a set of pre-defined substitution variables. If it is running in the lookup mode, the OID Migration Tool can automatically determine the values of these variables by looking them up Oracle Internet Directory.

Table I–4 Pre-defined Substitution Variables

Variable Name	Meaning	How OID Migration Tool Determines the Value for This Variable
%s_UserContainerDN%	Distinguished name of the entry under which all users are supposed to be added.	This is assigned the value of the attribute: orclCommonUserSearchBase from the entry cn=Common,cn=Products under the subscriber specific Oracle context.

Table I–4 Pre-defined Substitution Variables

Variable Name	Meaning	How OID Migration Tool Determines the Value for This Variable
%s_GroupContainerDN%	Distinguished name of the entry under which all public groups are supposed to be added.	This is assigned the value of the attribute: orclCommonGroupSearchBase from the entry cn=Common,cn=Products under the subscriber specific Oracle context.
%s_UserNicknameAttribute%	The nickname attribute to be used for user entries in the subscriber.	This is assigned the value of the attribute: orclCommonNicknameAttribute from the entry cn=Common,cn=Products under the subscriber specific Oracle context.
%s_SubscriberDN%	Distinguished name of the LDAP entry corresponding to the subscriber.	If a simple subscriber name is given, the migration tool will resolve it to a DN using the attribute: orclSubscriberSearchBase and the orclSubscriberNickNameAttr from the entry cn=Common,cn=Products under the root Oracle context.
%s_SubscriberOracleContextDN%	Distinguished name of the subscriber specific Oracle Context.	First the subscriber DN is computed as described above and then the string cn=OracleContext is pre-pended to it.
%s_RootOracleContextDN%	Distinguished name of the Root Oracle Context.	This is currently hard-coded to "cn=OracleContext".

Table I–4 Pre-defined Substitution Variables

Variable Name	Meaning	How OID Migration Tool Determines the Value for This Variable
%s_CurrentUserDN%	Distinguished name of the User who is loading the LDIF file. This is sometimes required to bootstrap the creation of groups which require at least one member in them.	The migration tool expects this DN to be specified on the command line as part of the authentication information.

The OID Migration Tool obtains the values of the pre-defined substitution variables only in the lookup mode. Users can override the value of any of the above variables in the 'lookup' mode by specifying the variable and a different value in the command line. The user can also specify substitution variables other than the ones listed in the table below and their values in the command line.

Examples: Using the OID Migration Tool

Consider the input file `sample.dat` whose contents are as follows:

```
dn: cn=jdoe, %s_UserContainerDN%
sn: Doe
%s_UserNicknameAttribute%: jdoe
objectClass: inetOrgPerson
objectClass: orclUserV2
title: Member of Technical Staff
homePhone: 415-584-5670
homePostalAddress: 234 Lez Drive$ Redwood City$ CA$ 94402
ou: %s_UserOrganization%
```

The following sections describe how the OID Migration Tool can be used to transform the above template into a valid LDIF ready to be loaded into Oracle Internet Directory.

Using the Migration Tool in the Lookup Mode

In this example, the Oracle directory server is present in the environment, and the deployment wants the migration tool to lookup the directory server to figure out certain substitution variables. It will issue the following command:

```
$ldifmigrator "input_file=sample.dat" "output_file=sample.ldif" -lookup
```

"host=ldap.acme.com" "subscriber=acme" "s_UserOrganization=Development"

On executing the above command, the directory server running on ldap.acme.com will be contacted and the following values of the substitution variables for the subscriber “acme” will be obtained:

Variable Name	Value Obtained from ldap.acme.com
% s_UserContainerDN%	cn=Users,o=acme,dc=com
%s_ UserNicknameAttribute %	uid

In addition to the above variables, the OID Migration Tool will also honor the command-line variable called s_UserOrganization and substitute all occurrences of it with the value ‘Development’. In this case the output of the tool stored in `sample.ldif` will be as follows (the substituted values are shown in italics):

```
dn: cn=jdoe,cn=Users,o=Acme,dc=com
sn: Doe
uid: jdoe
objectClass: inetOrgPerson
objectClass: orclUserV2
title: Member of Technical Staff
homePhone: 415-584-5670
homePostalAddress: 234 Lez Drive$ Redwood City$ CA$ 94402
ou: Development
```

Using the OID Migration Tool Without the Lookup Option

The same output as shown in the previous example could have been obtained by specifying all of the values in the command line (without using the -lookup option). The following command line example describes how one would use the Migration tool without the lookup mode:

```
$ldifmigrator "input_file=sample.dat" "output_file=sample.ldif"  "s_
UserContainerDN=cn=Users,o=Acme,dc=com" "s_UserNicknameAttribute=uid" "s_
UserOrganization=Development"
```

Overriding Substitution Values Obtained from the Lookup Mode

In some cases, a deployment would like to use the OID Migration Tool in the lookup mode but would also like to override the values of one or more of the pre-defined substitution variables. This can be done by specifying the override

value in the command line. The following command line shows how one can set the UserNickNameAttribute to 'cn' overriding the default of 'uid':

```
$ldifmigrator "input_file=sample.dat" "output_file=sample.ldif" -lookup
"host=ldap.acme.com" "subscriber=acme" "s_UserOrganization=Development"
"s_UserNicknameAttribute=cn"
```

On executing the above command, the directory server running on ldap.acme.com will be contacted and the following values of the substitution variables for the subscriber "acme" will be obtained:

Variable Name	Value Obtained from ldap.acme.com
% s_UserContainerDN%	cn=Users,o=acme,dc=com
%s_UserNicknameAttribute%	uid (this is over-ridden by command line specification)

Since `s_UserNicknameAttribute` is specified on the command line, the OID Migration Tool will ignore the value obtained from the directory and use the value specified in the command line. In addition to the above variables, the migration tool will also honor the command-line variable called `s_UserOrganization` and substitute all occurrences of it with the value 'Development'. In this case the output of the tool stored in `sample.ldif` will be as follows (the substituted values are shown in italics):

```
dn: cn=jdoe,cn=Users,o=Acme,dc=com
sn: Doe
cn: jdoe
objectClass: inetOrgPerson
objectClass: orclUserV2
title: Member of Technical Staff
homePhone: 415-584-5670
homePostalAddress: 234 Lez Drive$ Redwood City$ CA$ 94402
ou: Development
```

OID Migration Tool Error Messages

The OID Migration Tool can display these error messages:

Message	Reason	Remedial Action
Environment variable *ORACLE_HOME* not defined	*ORACLE_HOME* is not defined.	Set the environment variable *ORACLE_HOME*

Message	Reason	Remedial Action
Error while parsing the input parameters. Please verify	Not all the required parameters are provided. The required parameters are Input_File, Output_File and at least one substitution variable	Specify the input parameters properly. Use the `-help` option to print the usage.
Input_File parameter not specified. Please specify	Input_File parameter is a mandatory parameter.	Specify the input parameters properly. Use the `-help` option to print the usage.
Output_File parameter not specified. Please specify	Output_File parameter is a mandatory parameter.	Specify the input parameters properly. Use the `-help` option to print the usage.
The specified input file does not exist	The specified file location is invalid.	Check the input file path
Check the input file. Zero byte input file	The input file does not contain any entries.	Provide a valid file with pseudo LDIF entries
Cannot create the output file. Output file already exists	The output file already exists	Check the Output_File flag
Access denied, cannot read from the input file	The specified input file does not have read permission	Check the read permission of the input file.
Access denied, cannot create the output file	You do not have permission to create the output file.	Check the permission of the directory under which the output file needs to be created.
Directory server name not specified. When -lookup option is used the host parameter should be specified	When the `-lookup` option is specified, the host parameter is mandatory.	Specify the host parameter.
Bind Dn parameter name not specified. When -lookup option is used the dn parameter should be specified	When the `-lookup` option is specified, the DN parameter is mandatory.	Specify the DN parameter.
The port number specified is invalid	The port number should be a numeric value.	Check the port number parameter

Message	Reason	Remedial Action
Unable to establish connection to directory. Please verify the input parameters: host, port, dn & password	The directory server may not be running on the specified host and port, or credentials may be invalid.	Check the host, port, DN and password parameters. Check $*ORACLE_HOME*/ldap/install/LDIFMig_YYYY_MM_DD_HH_SS.log file.
Naming Exception occurred while retrieving the subscriber information from the directory. Please verify the input parameters	The specified subscriber does not exist in the directory	Check the subscriber parameter
Not all the substitution variables are defined in the directory server specified	If the subscriber entry does not contain the required attributes, then this error occurs.	Check the subscriber entry in the directory
Error occurred while migrating LDIF data to OID	This might occur if something goes wrong in the middle of a process—for example, a failure of the directory server or disk.	Report the error message to the administrator

When an error condition occurs, the log messages are logged to this file: *ORACLE_HOME*/ldap/install/LDIFMig_YYYY_MM_DD_HH_SS.log.

Glossary

access control item (ACI)

An attribute that determines who has what type of access to what directory data. It contains a set of rules for structural access items, which pertain to entries, and content access items, which pertain to attributes. Access to both structural and content access items may be granted to one or more users or groups.

access control list (ACL)

The group of access directives that you define. The directives grant levels of access to specific data for specific clients, or groups of clients, or both.

access control policy point

An entry that contains security directives that apply downward to all entries at lower positions in the **directory information tree (DIT)**.

ACI

See **access control item (ACI)**.

ACL

See **access control list (ACL)**.

ACP

See **access control policy point**.

administrative area

A subtree on a directory server whose entries are under the control (schema, ACL, and collective attributes) of a single administrative authority.

advanced symmetric replication (ASR)

See **Oracle9i Replication**

agent

See **directory integration agent**

agent profile

In an Oracle Directory Integration Platform environment, an entry in Oracle Internet Directory that specifies:

- Configuration parameters for integration agents
- Mapping rules for synchronizing between a connected directory and Oracle Internet Directory

anonymous authentication

The process by which the directory authenticates a user without requiring a user name and password combination. Each anonymous user then exercises the privileges specified for anonymous users.

API

See **application program interface**.

application program interface

Programs to access the services of a specified application. For example, LDAP-enabled clients access directory information through programmatic calls available in the LDAP API.

ASR

See **Oracle9i Replication**

attribute

An item of information that describes some aspect of an entry. An entry comprises a set of attributes, each of which belongs to an **object class**. Moreover, each attribute has both a *type*, which describes the kind of information in the attribute, and a *value*, which contains the actual data.

attribute configuration file

In an Oracle Directory Integration Platform environment, a file that specifies attributes of interest in a connected directory.

attribute type

The kind of information an attribute contains, for example, `jobTitle`.

attribute value

The particular occurrence of information appearing in that entry. For example, the value for the `jobTitle` attribute could be `manager`.

authentication

The process of verifying the identity of a user, device, or other entity in a computer system, often as a prerequisite to allowing access to resources in a system.

authorization

Permission given to a user, program, or process to access an object or set of objects.

binding

The process of authenticating to a directory.

central directory

In an Oracle Directory Integration Platform environment, the directory that acts as the central repository. In an Oracle Directory Integration Platform environment, Oracle Internet Directory is the central directory.

certificate

An ITU x.509 v3 standard data structure that securely binds an identity to a public key. A certificate is created when an entity's public key is signed by a trusted identity: a **certificate authority (CA)**. This certificate ensures that the entity's information is correct and that the public key actually belongs to that entity.

certificate authority (CA)

A trusted third party that certifies that other entities—users, databases, administrators, clients, servers—are who they say they are. The certificate authority verifies the user's identity and grants a certificate, signing it with the certificate authority's private key.

certificate chain

An ordered list of certificates containing an end-user or subscriber certificate and its certificate authority certificates.

change logs

A database that records changes made to a directory server.

cipher suite

In SSL, a set of authentication, encryption, and data integrity algorithms used for exchanging messages between network nodes. During an SSL handshake, the two nodes negotiate to see which cipher suite they will use when transmitting messages back and forth.

cold backup

The procedure to add a new **DSA** node to an existing replicating system by using the database copy procedure.

concurrency

The ability to handle multiple requests simultaneously. Threads and processes are examples of concurrency mechanisms.

concurrent clients

The total number of clients that have established a session with Oracle Internet Directory.

concurrent operations

The number of operations that are being executed on the directory from all of the concurrent clients. Note that this is not necessarily the same as the concurrent clients, because some of the clients may be keeping their sessions idle.

configset

See **configuration set entry**.

configuration set entry

A directory entry holding the configuration parameters for a specific instance of the directory server. Multiple configuration set entries can be stored and referenced at run-time. The configuration set entries are maintained in the subtree specified by the subConfigsubEntry attribute of the DSE, which itself resides in the associated **directory information base (DIB)** against which the servers are started.

connect descriptor

A specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information.

The destination service is indicated by using its service name for Oracle9*i* release 9.0.1 database or its Oracle System Identifier (SID) for Oracle release 8.0 or version 7 databases. The network route provides, at a minimum, the location of the listener through use of a network address.

connected directory

In an Oracle Directory Integration Platform environment, an information repository requiring full synchronization of data between Oracle Internet Directory and itself—for example, an Oracle human Resources database.

consumer

A directory server that is the destination of replication updates. Sometimes called a slave.

contention

Competition for resources.

context prefix

The **DN** of the root of a **naming context**.

cryptography

The practice of encoding and decoding data, resulting in secure messages.

data integrity

The guarantee that the contents of the message received were not altered from the contents of the original message sent.

decryption

The process of converting the contents of an encrypted message (ciphertext) back into its original readable format (plaintext).

default knowledge reference

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

DES

Data Encryption Standard, a block cipher developed by IBM and the U.S. government in the 1970's as an official standard.

DIB

See **directory information base (DIB)**.

directory information base (DIB)

The complete set of all information held in the directory. The DIB consists of entries that are related to each other hierarchically in a **directory information tree (DIT).**

directory information tree (DIT)

A hierarchical tree-like structure consisting of the DNs of the entries.

directory integration agent

In an Oracle Directory Integration Platform environment, a program that interacts with a connected directory to synchronize changes between the connected directory and Oracle Internet Directory.

directory integration profile

In an Oracle Directory Integration Platform environment, an entry in Oracle Internet Directory that describes how Oracle Directory Integration Platform communicates with external systems and what is communicated.

directory integration server

In an Oracle Directory Integration Platform environment, the server that drives the synchronization of data between Oracle Internet Directory and a **connected directory**.

directory naming context

See **naming context**.

Directory Provisioning Profile

A special kind of **directory integration profile** that describes the nature of provisioning-related notifications that the Oracle Directory Integration Platform sends to the directory-enabled applications

directory replication group (DRG)

The directory servers participating in a replication agreement.

directory server instance

A discrete invocation of a directory server. Different invocations of a directory server, each started with the same or different configuration set entries and startup flags, are said to be different directory server instances.

directory-specific entry (DSE)

An entry specific to a directory server. Different directory servers may hold the same DIT name, but have different contents—that is, the contents can be specific to the directory holding it. A DSE is an entry with contents specific to the directory server holding it.

directory synchronization profile

A special kind of **directory integration profile** that describes how synchronization is carried out between Oracle Internet Directory and an external system.

directory system agent (DSA)

The X.500 term for a directory server.

distinguished name (DN)

The unique name of a directory entry. It comprises all of the individual names of the parent entries back to the root.

DIS

See **directory integration server**

DIT

See **directory information tree (DIT)**

DN

See **distinguished name (DN)**

DRG

See **directory replication group (DRG)**

DSA

See **directory system agent (DSA)**

DSE

See **directory-specific entry (DSE)**

DSA-specific entries. Different DSAs may hold the same DIT name, but have different contents. That is, the contents can be specific to the DSA holding it. A DSE is an entry with contents specific to the DSA holding it.

encryption

The process of disguising the contents of a message and rendering it unreadable (ciphertext) to anyone but the intended recipient.

entry

The building block of a directory, it contains information about an object of interest to directory users.

export agent

In an Oracle Directory Integration Platform environment, an agent that exports data out of Oracle Internet Directory.

export data file

In an Oracle Directory Integration Platform environment, the file that contains data exported by an **export agent**.

export file

See **export data file**.

external agent

A directory integration agent that is independent of Oracle directory integration server. The Oracle directory integration server does not provide scheduling, mapping, or error handling services for it. An external agent is typically used when a third party metadirectory solution is integrated with the Oracle Directory Integration Platform.

failover

The process of failure recognition and recovery.

filter

A method of qualifying data, usually data that you are seeking. Filters are always expressed as DNs, for example: `cn=susie smith, o=acme, c=us`.

global unique identifier (GUID)

In a multi-master replication environment, an entry replicated on multiple nodes has the same DN on each node. However, even though it has the same DN, it is

assigned a different GUID on each node. For example, the same DN can be replicated on both node1 and node2, but the GUID for that DN as it resides on node1 would be different from the GUID for that DN on node2.

grace login

A login occurring within the specified period before password expiration.

guest user

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

GUID

See **global unique identifier (GUID)**.

handshake

A protocol two computers use to initiate a communication session.

hash

A number generated from a string of text with an algorithm. The hash value is substantially smaller than the text itself. Hash numbers are used for security and for faster access to data.

import agent

In an Oracle Directory Integration Platform environment, an agent that imports data into Oracle Internet Directory.

import data file

In an Oracle Directory Integration Platform environment, the file containing the data imported by an **import agent**.

inherit

When an object class has been derived from another class, it also derives, or inherits, many of the characteristics of that other class. Similarly, an attribute subtype inherits the characteristics of its supertype.

instance

See **directory server instance**.

integration agent

See **agent**.

integrity

The guarantee that the contents of the message received were not altered from the contents of the original message sent.

Internet Engineering Task Force (IETF)

The principal body engaged in the development of new Internet standard specifications. It is an international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet.

Internet Message Access Protocol (IMAP)

A protocol allowing a client to access and manipulate electronic mail messages on a server. It permits manipulation of remote message folders, also called mailboxes, in a way that is functionally equivalent to local mailboxes.

key

A string of bits used widely in cryptography, allowing people to encrypt and decrypt data; a key can be used to perform other mathematical operations as well. Given a cipher, a key determines the mapping of the plaintext to the ciphertext.

key pair

A **public key** and its associated **private key**.

See **public/private key pair**.

knowledge reference

The access information (name and address) for a remote **DSA** and the name of the **DIT** subtree that the remote DSA holds. Knowledge references are also called referrals.

latency

The time a client has to wait for a given directory operation to complete. Latency can be defined as wasted time. In networking discussions, latency is defined as the travel time of a packet from source to destination.

LDAP

See **Lightweight Directory Access Protocol (LDAP)**.

LDIF

See **LDAP Data Interchange Format (LDIF)**.

Lightweight Directory Access Protocol (LDAP)

A standard, extensible directory access protocol. It is a common language that LDAP clients and servers use to communicate. The framework of design conventions supporting industry-standard directory products, such as the Oracle Internet Directory.

LDAP Data Interchange Format (LDIF)

The set of standards for formatting an input file for any of the LDAP command-line utilities.

man-in-the-middle

A security attack characterized by the third-party, surreptitious interception of a message. The third-party, the *man-in-the-middle*, decrypts the message, re-encrypts it (with or without alteration of the original message), and retransmits it to the originally-intended recipient—all without the knowledge of the legitimate sender and receiver. This type of security attack works only in the absence of **authentication**.

mapping rules file

In an Oracle Directory Integration Platform environment, the file that specifies mappings between Oracle Internet Directory attributes and those in a **connected directory**.

master definition site (MDS)

In replication, a master definition site is the Oracle Internet Directory database from which the administrator runs the configuration scripts.

master site

In replication, a master site is any site other than the master definition site that participates in LDAP replication.

matching rule

In a search or compare operation, determines equality between the attribute value sought and the attribute value stored. For example, matching rules associated with the `telephoneNumber` attribute could cause "(650) 123-4567" to be matched with either "(650) 123-4567" or "6501234567" or both. When you create an attribute, you associate a matching rule with it.

MD4

A one-way hash function that produces a 128-bit hash, or message digest. If as little as a single bit value in the file is modified, the MD4 checksum for the file will change. Forgery of a file in a way that will cause MD4 to generate the same result as that for the original file is considered extremely difficult.

MD5

An improved version of MD4.

MDS

See **master definition site (MDS)**

metadirectory

A directory solution that shares information between all enterprise directories, integrating them into one virtual directory. It centralizes administration, thereby reducing administrative costs. It synchronizes data between directories, thereby ensuring that it is consistent and up-to-date across the enterprise.

MTS

See **shared server**

native agent

In an Oracle Directory Integration Platform environment, an **agent** that runs under the control of the **directory integration server**.

naming attribute

A specialized attribute that holds values for different types of **RDN**. A naming attribute is identifiable by its mnemonic label, usually `cn`, `sn`, `ou`, `o`, `c`, and so on. For example, the naming attribute `c` is the mnemonic for the naming attribute `country`, and it holds the RDN for specific country values.

naming context

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

Oracle Net Services

The foundation of the Oracle family of networking products, allowing services and their client applications to reside on different computers and communicate. The

main function of Oracle Net Services is to establish network sessions and transfer data between a client application and a server. Oracle Net Services is located on each computer in the network. Once a network session is established, Oracle Net Services acts as a data courier for the client and the server.

net service name

A simple name for a service that resolves to a connect descriptor. Users initiate a connect request by passing a user name and password along with a net service name in a connect string for the service to which they wish to connect:

```
CONNECT username/password@net_service_name
```

Depending on your needs, net service names can be stored in a variety of places, including:

- Local configuration file, tnsnames.ora, on each client
- Directory server
- Oracle Names server
- External naming service, such as NDS, NIS or CDS

object class

A named group of attributes. When you want to assign attributes to an entry, you do so by assigning to that entry the object classes that hold those attributes.

All objects associated with the same object class share the same attributes.

OEM

See **Oracle Enterprise Manager**.

OID Control Utility

A command-line tool for issuing run-server and stop-server commands. The commands are interpreted and executed by the **OID Monitor** process.

OID Database Password Utility

The utility used to change the password with which Oracle Internet Directory connects to an Oracle database.

OID Monitor

The Oracle Internet Directory component that initiates, monitors, and terminates the Oracle directory server processes. It also controls the replication server if one is installed, and Oracle directory integration server.

one-way function

A function that is easy to compute in one direction but quite difficult to reverse compute, that is, to compute in the opposite direction.

one-way hash function

A **one-way function** that takes a variable sized input and creates a fixed size output.

Oracle Call Interface (OCI)

An application programming interface (API) that allows you to create applications that use the native procedures or function calls of a third-generation language to access an Oracle database server and control all phases of SQL statement execution.

Oracle Directory Integration Platform

A component of **Oracle Internet Directory**. It is a framework developed to integrate applications around a central LDAP directory like Oracle Internet Directory.

Oracle directory integration server (DIS)

In an Oracle Directory Integration Platform environment, a daemon process that monitors Oracle Internet Directory for change events and takes action based on the information present in the **directory integration profile**.

Oracle Directory Manager

A Java-based tool with a graphical user interface for administering Oracle Internet Directory.

Oracle Enterprise Manager

A separate Oracle product that combines a graphical console, agents, common services, and tools to provide an integrated and comprehensive systems management platform for managing Oracle products.

Oracle Internet Directory

A general purpose directory service that enables retrieval of information about dispersed users and network resources. It combines Lightweight Directory Access

Protocol (LDAP) Version 3 with the high performance, scalability, robustness, and availability of Oracle9*i*.

Oracle PKI certificate usages

Defines Oracle application types that a **certificate** supports.

Oracle Wallet Manager

A Java-based application that security administrators use to manage public-key security credentials on clients and servers.

Oracle9*i* Replication

A feature in Oracle9*i* that allows database tables to be kept synchronized across two Oracle databases.

other information repository

In an Oracle Directory Integration Platform environment, in which Oracle Internet Directory serves as the **central directory**, any information repository except Oracle Internet Directory.

partition

A unique, non-overlapping directory naming context that is stored on one directory server.

partner agent

A directory integration agent for which Oracle directory integration server performs mapping, scheduling, and error handling.

PKCS #12

A **public-key encryption** standard (PKCS). RSA Data Security, Inc. PKCS #12 is an industry standard for storing and transferring personal authentication credentials—typically in a format called a **wallet**.

plaintext

Message text that has not been encrypted.

private key

In public-key cryptography, this key is the secret key. It is primarily used for decryption but is also used for encryption with digital signatures.

provisioning agent

An application or process that translates Oracle-specific provisioning events to external or third-party application-specific events.

provisioned applications

Applications in an environment where user and group information is centralized in Oracle Internet Directory. These applications are typically interested in changes to that information in Oracle Internet Directory.

profile

See **directory integration profile**

proxy user

A kind of user typically employed in an environment with a middle tier such as a firewall. In such an environment, the end user authenticates to the middle tier. The middle tier then logs into the directory on the end user's behalf. A proxy user has the privilege to switch identities and, once it has logged into the directory, switches to the end user's identity. It then performs operations on the end user's behalf, using the authorization appropriate to that particular end user.

public key

In public-key cryptography this key is made public to all, it is primarily used for encryption but can be used for verifying signatures.

public-key cryptography

Cryptography based on methods involving a public key and a private key.

public-key encryption

The process in which the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the message is decrypted by the recipient using the recipient's private key.

public/private key pair

A mathematically related set of two numbers where one is called the private key and the other is called the public key. Public keys are typically made widely available, while private keys are available only to their owners. Data encrypted with a public key can only be decrypted with its associated private key and vice versa. Data encrypted with a public key cannot be decrypted with the same public key.

referral

Information that a directory server provides to a client and which points to other servers the client must contact to find the information it is requesting.

See also **knowledge reference**.

relational database

A structured collection of data that stores data in tables consisting of one or more rows, each containing the same set of columns. Oracle makes it very easy to link the data in multiple tables. This is what makes Oracle a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the tables. The link is based on one or more fields common to both tables.

replica

Each copy of a naming context that is contained within a single server.

RDN

See **relative distinguished name (RDN)**.

registry entry

An entry containing runtime information associated with invocations of Oracle directory servers, called a **directory server instance**. 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`.

remote master site (RMS)

In a replicated environment, any site, other than the **master definition site (MDS)**, that participates in Oracle9*i* Replication.

replication agreement

A special directory entry that represents the replication relationship among the directory servers in a **directory replication group (DRG)**.

response time

The time between the submission of a request and the completion of the response.

root DSE

See **root directory specific entry**.

root directory specific entry

An entry storing operational information about the directory. The information is stored in a number of attributes.

SASL

See **Simple Authentication and Security Layer (SASL)**

scalability

The ability of a system to provide throughput in proportion to, and limited only by, available hardware resources.

schema

The collection of attributes, object classes, and their corresponding matching rules.

Secure Hash Algorithm (SHA)

An algorithm that takes a message of less than 264 bits in length and produces a 160-bit message digest. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.

Secure Socket Layer (SSL)

An industry standard protocol designed by Netscape Communications Corporation for securing network connections. SSL provides authentication, encryption, and data integrity using public key infrastructure (PKI).

service time

The time between the initiation of a request and the completion of the response to the request.

session key

A key for symmetric-key cryptosystems that is used for the duration of one message or communication session.

SGA

See **System Global Area (SGA)**.

SHA

See **Secure Hash Algorithm (SHA)**.

shared server

A server that is configured to allow many user processes to share very few server processes, so the number of users that can be supported is increased. With shared server configuration, many user processes connect to a dispatcher. The dispatcher directs multiple incoming network session requests to a common queue. An idle shared server process from a shared pool of server processes picks up a request from the queue. This means a small pool of server processes can server a large amount of clients. Contrast with dedicated server.

sibling

An entry that has the same parent as one or more other entries.

simple authentication

The process by which the client identifies itself to the server by means of a DN and a password which are not encrypted when sent over the network. In the simple authentication option, the server verifies that the DN and password sent by the client match the DN and password stored in the directory.

Simple Authentication and Security Layer (SASL)

A method for adding authentication support to connection-based protocols. To use this specification, a protocol includes a command for identifying and authenticating a user to a server and for optionally negotiating a security layer for subsequent protocol interactions. The command has a required argument identifying a SASL mechanism.

single key-pair wallet

A **PKCS #12**-format **wallet** that contains a single user **certificate** and its associated **private key**. The **public key** is imbedded in the certificate.

slave

See **consumer**.

SLAPD

Standalone LDAP daemon.

smart knowledge reference

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

specific administrative area

Administrative areas control:

- Subschema administration
- Access control administration
- Collective attribute administration

A *specific* administrative area controls one of the above aspects of administration. A specific administrative area is part of an autonomous administrative area.

sponsor node

In replication, the node that is used to provide initial data to a new node.

SSL

See **Secure Socket Layer (SSL)**.

subclass

An object class derived from another object class. The object class from which it is derived is called its **superclass**.

subschema DN

The list of DIT areas having independent schema definitions.

subentry

A type of entry containing information applicable to a group of entries in a subtree. The information can be of these types:

- Access control policy points
- Schema rules
- Collective attributes

Subentries are located immediately below the root of an administrative area.

subordinate reference

A knowledge reference pointing downward in the DIT to a naming context that starts immediately below an entry.

subtype

An attribute with one or more options, in contrast to that same attribute without the options. For example, a commonName (cn) attribute with American English as an option is a subtype of the commonName (cn) attribute without that option. Conversely, the commonName (cn) attribute without an option is the **supertype** of the same attribute with an option.

subACLSubentry

A specific type of subentry that contains ACL information.

subSchemaSubentry

A specific type of **subentry** containing schema information.

super user

A special directory administrator who typically has full access to directory information.

superclass

The object class from which another object class is derived. For example, the object class person is the superclass of the object class organizationalPerson. The latter, namely, organizationalPerson, is a **subclass** of person and inherits the attributes contained in person.

superior reference

A knowledge reference pointing upward to a DSA that holds a naming context higher in the DIT than all the naming contexts held by the referencing DSA.

supertype

An attribute without options, in contrast to the same attribute with one or more options. For example, the commonName (cn) attribute without an option is the supertype of the same attribute with an option. Conversely, a commonName (cn) attribute with American English as an option is a **subtype** of the commonName (cn) attribute without that option.

supplier

In replication, the server that holds the master copy of the naming context. It supplies updates from the master copy to the **consumer** server.

System Global Area (SGA)

A group of shared memory structures that contain data and control information for one Oracle database instance. If multiple users are concurrently connected to the same instance, the data in the instance SGA is shared among the users. Consequently, the SGA is sometimes referred to as the "shared global area." The combination of the background processes and memory buffers is called an Oracle instance.

system operational attribute

An attribute holding information that pertains to the operation of the directory itself. Some operational information is specified by the directory to control the server, for example, the time stamp for an entry. Other operational information, such as access information, is defined by administrators and is used by the directory program in its processing.

TLS

See **Transport Layer Security (TLS)**

think time

The time the user is not engaged in actual use of the processor.

throughput

The number of requests processed by Oracle Internet Directory per unit of time. This is typically represented as "operations per second."

Transport Layer Security (TLS)

A protocol providing communications privacy over the Internet. The protocol allows client-server applications to communicate in a way that prevents eavesdropping, tampering, or message forgery.

trusted certificate

A third party identity that is qualified with a level of trust. The trust is used when an identity is being validated as the entity it claims to be. Typically, the certificate authorities you trust issue user certificates.

trustpoint

See **trusted certificate**.

UCS-2

Fixed-width 16-bit **Unicode**. Each character occupies 16 bits of storage. The Latin-1 characters are the first 256 code points in this standard, so it can be viewed as a 16-bit extension of Latin-1.

Unicode

A type of universal character set, a collection of 64K characters encoded in a 16-bit space. It encodes nearly every character in just about every existing character set standard, covering most written scripts used in the world. It is owned and defined by Unicode Inc. Unicode is canonical encoding which means its value can be passed around in different locales. But it does not guarantee a round-trip conversion between it and every Oracle character set without information loss.

UNIX Crypt

The UNIX encryption algorithm.

UTC (Coordinated Universal Time)

The standard time common to every place in the world. Formerly and still widely called Greenwich Mean Time (GMT) and also World Time, UTC nominally reflects the mean solar time along the Earth's prime meridian. UTC is indicated by a z at the end of the value, for example, 200011281010z.

UTF-8

A variable-width encoding of **UCS-2** which uses sequences of 1, 2, or 3 bytes per character. Characters from 0-127 (the 7-bit ASCII characters) are encoded with one byte, characters from 128-2047 require two bytes, and characters from 2048-65535 require three bytes. The Oracle character set name for this is UTF-8 (for the Unicode 2.1 standard). The standard has left room for expansion to support the UCS4 characters with sequences of 4, 5, and 6 bytes per character.

wallet

An abstraction used to store and manage security credentials for an individual entity. It implements the storage and retrieval of credentials for use with various cryptographic services. A wallet resource locator (WRL) provides all the necessary information to locate the wallet.

wait time

The time between the submission of the request and initiation of the response.

X.509

A popular format from ISO used to sign public keys.

Index

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W