Oracle® Internet Directory

Administrator's Guide Volume 1

10*g* (9.0.4)

Part No. B12118-01

September 2003

ORACLE®

Oracle Internet Directory Administrator's Guide, 10*g* (9.0.4)

Part No. B12118-01

Primary Author: Richard Smith

Contributing Authors: Jennifer Polk

Contributors: Vasuki Ashok, Tridip Bhattacharya, Neelima Bawa, Ramakrishna Bollu, Margaret Chou, Saheli Dey, Rajinder Gupta, Ajay Keni, Ashish Kolli, Stephen Lee, David Lin, Michael Mesaros, Radhika Moolky, Hari Sastry, David Saslav, Ramaprakash Sathyanarayan, Bhupindra Singh, Gurudatt Shakshikumar, Amit Sharma, Jason Sharma, Daniel Shih, Saurabh Shrivastava, Uppili Srinivasan, Olaf Stullich, Dipankar Thakuria, Sivakumar Venugopal

Contents

2 Directory Concepts and Architecture

6 Directory Schema Administration

7 Directory Entries Administration

8 Attribute Uniqueness in the Directory

9 Dynamic and Static Groups in Oracle Internet Directory

12 Directory Security Concepts

13 Secure Sockets Layer (SSL) and the Directory

14 Directory Access Control

15 Password Policies in Oracle Internet Directory

16 Directory Storage of Password Verifiers

17 Delegation of Privileges for an Oracle Technology Deployment

Part IV Directory Deployment

18 Directory Deployment Considerations

19 Deployment of Oracle Identity Management Realms

20 Capacity Planning for the Directory

21 Tuning Considerations for the Directory

22 Garbage Collection in Oracle Internet Directory

23 Migration of Data from Other Directories

Part V Directory Replication and High Availability

24 Directory Replication Concepts

25 Oracle Directory Replication Administration

26 High Availability And Failover Considerations

27 Rack-Mounted Directory Server Configurations

28 Cold Failover Cluster Configuration

29 The Directory in an Oracle9*i* Real Application Clusters Environment

Volume 2

Part VI Delegation and Self-Service Administration in Oracle Internet Directory

30 Oracle Delegated Administration Services

31 Oracle Internet Directory Self-Service Console

Part VII Oracle Directory Integration and Provisioning Platform

32 Oracle Directory Integration and Provisioning Platform Concepts and Components

33 Oracle Directory Synchronization Service

34 Oracle Directory Provisioning Integration Service

35 Oracle Directory Integration and Provisioning Server Administration

36 Security in the Oracle Directory Integration and Provisioning Platform

37 Bootstrapping of a Directory in the Oracle Directory Integration and Provisioning Platform

38 Synchronization with Relational Database Tables

39 Synchronization with Oracle Human Resources

43 Integration with the Microsoft Windows Environment

47 Setting Up the Customized External Authentication Plug-in

Part IX Appendixes

A Syntax for LDIF and Command-Line Tools

B Oracle Internet Directory Schema Elements

C Elements in Oracle Internet Directory Graphical User Interfaces

D The LDAP Filter Definition

E The Access Control Directive Format

F Addition of a Directory Node by Using the Database Copy Procedure

G Globalization Support in the Directory

H Troubleshooting

List of Figures

List of Tables

Send Us Your Comments

Oracle Internet Directory Administrator's Guide, 10*g* (9.0.4)

Part No. B12118-01

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: appserverdocs@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.

- For information about routine administration:
  - Part I: Getting Started
  - Part II: Basic Directory Administration
- For information about directory planning and deployment in enterprises and hosted environments:
  - Part III Directory Security
  - Part IV Directory Deployment
  - Part V Directory Replication and High Availability
  - Part VI: Delegation and Self-Service Administration in Oracle Internet Directory
- For information about integration between Oracle Internet Directory and other directories, see Part VII: The Oracle Directory Integration and Provisioning Platform
- For information about extending Oracle Internet Directory functionality by using plug-ins, see Part VIII: Oracle Internet Directory Plug-ins

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 to LDAP and Oracle Internet Directory"

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

Chapter 2, "Directory 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 Globalization 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, "Directory Entries Administration"

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

Chapter 8, "Attribute Uniqueness in the Directory"

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.

Chapter 9, "Dynamic and Static Groups in Oracle Internet Directory"

This chapter describes both static and dynamic groups and explains how to administer them in Oracle Internet Directory.

Chapter 10, "Logging, Auditing, and Monitoring the Directory"

This chapter describes the comprehensive framework provided by Oracle Internet Directory for enabling you to debug, audit, and monitor the directory.s

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, "Backup and Restoration of a Directory"

This appendix tells how to backup and restore both small and large directories.

Chapter 12, "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 13, "Secure Sockets Layer (SSL) and the Directory"

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

Chapter 14, "Directory Access Control"

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

Chapter 15, "Password Policies in Oracle Internet Directory"

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 16, "Directory Storage of Password Verifiers"

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 17, "Delegation of Privileges for an Oracle Technology Deployment"

This chapter explains how to store all the data for users, groups, and services in one repository, and delegate the administration of that data to various administrators. It also explains the default security configuration in Oracle Internet Directory.

Part IV Directory Deployment

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

Chapter 18, "Directory 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 19, "Deployment of Oracle Identity Management Realms"

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 20, "Capacity Planning for the Directory"

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 21, "Tuning Considerations for the Directory"

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

Chapter 22, "Garbage Collection in Oracle Internet Directory"

The term "garbage" refers to any data not needed by the directory but still occupying space on it. The process of removing this unwanted data from the directory is called garbage collection. This chapter describes the predefined garbage collectors available with Oracle Internet Directory, and tells how to modify them.

Chapter 23, "Migration of Data from Other Directories"

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

Part V Directory Replication and High Availability

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

Chapter 24, "Directory Replication Concepts"

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

Chapter 25, "Oracle Directory Replication 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 26, "High Availability And Failover Considerations"

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

Chapter 27, "Rack-Mounted Directory Server Configurations"

This chapter describes rack-mounted directory server configuration, which provides high availability of a directory server. This configuration involves running multiple directory server instances on different hardware nodes. The directory servers are connected to the same directory store, which is an Oracle9*i* Database Server.

Chapter 28, "Cold Failover Cluster Configuration"

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

Chapter 29, "The Directory 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 VI: Delegation and Self-Service Administration in Oracle Internet Directory

Chapter 30, "Oracle Delegated Administration Services"

This chapter describes Oracle Delegated Administration Services, a framework consisting of pre-defined, Web-based units for building administrative and self-service consoles. These consoles can be used by Delegated administrators and users to perform specified directory operations.

Chapter 31, "Oracle Internet Directory Self-Service Console"

This chapter describes the Oracle Internet Directory Self-Service Console, a ready-to-use application created by using Oracle Delegated Administration Services.

Part VII: The Oracle Directory Integration and Provisioning Platform

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

Chapter 32, "Oracle Directory Integration and Provisioning Platform Concepts and Components"

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

Chapter 33, "Oracle Directory Synchronization Service"

This chapter discusses the synchronization profiles and connectors that link Oracle Internet Directory and connected directories.

Chapter 34, "Oracle Directory Provisioning Integration Service"

This chapter describes the Oracle Directory Provisioning Integration Service, which enables your applications to receive provisioning information from Oracle Internet Directory.

Chapter 35, "Oracle Directory Integration and Provisioning Server Administration"

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

Chapter 36, "Security in the Oracle Directory Integration and Provisioning Platform"

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

Chapter 37, "Bootstrapping of a Directory in the Oracle Directory Integration and Provisioning Platform"

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

Chapter 38, "Synchronization with Relational Database Tables"

This chapter explains how to synchronize data to Oracle Internet Directory from tables in a relational database. The synchronization can be either incremental—for example, one database table row at a time—or all the database tables at once.

Chapter 39, "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. This chapter explains the Oracle Human Resources agent, which enables you to do this.

Chapter 40, "Integration of Provisioning Data with the Oracle E-Business Suite"

In Oracle Internet Directory 10*g* (9.0.4), you can use the Oracle Directory Provisioning Integration Service to synchronize user accounts and other user information from the Oracle E-Business Suite.

Chapter 41, "Considerations for Integrating with Third-Party Directories"

Before you begin integrating any third-party directory with Oracle Internet Directory, you need to decide how you want to configure the integrated environment. This chapter discusses the basic decisions you need to make. Once you have made them, you can follow the steps for setting up successive bootstrapping and synchronization of data between the directories.

Chapter 42, "Integration with SunONE (iPlanet) Directory Server"

This chapter explains how you can synchronize between Oracle Internet Directory and an SunONE Directory Server by using the SunONE connector.

Chapter 43, "Integration with the Microsoft Windows Environment"

This chapter explains how to integrate the Oracle Application Server infrastructure with the Microsoft Windows Operating System. This integration is achieved by using the Active Directory Connector in the Oracle Directory Integration and Provisioning platform.

Chapter 44, "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.

Part VIII: Oracle Internet Directory Plug-ins

Chapter 45, "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.

Chapter 46, "Oracle Internet Directory Plug-In for Password Policies"

Oracle Internet Directory uses plug-ins to add password value checking to its other password policy management capabilities. These plug-ins enable you to verify that, for example, a new or modified password has the specified minimum length. You can customize password value checking to meet your own requirements. This chapter describes the plug-in for password policies and provides an example of its use.

Chapter 47, "Setting Up the Customized External Authentication Plug-in"

You can store user security credentials in a repository other than Oracle Internet Directory—for example, a database or another LDAP directory—and use these credentials for user authentication to Oracle components. You do not need to store the credentials in Oracle Internet Directory and then worry about keeping them synchronized. Authenticating a user by way of credentials stored in an external repository is called external authentication. This chapter describes the external authentication plug-in and provides an example of its use.

Part IX: Appendixes

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

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

Appendix B, "Oracle Internet Directory Schema Elements"

This appendix lists schema elements supported in Oracle Internet Directory.

Appendix C, "Elements in Oracle Internet Directory Graphical User Interfaces"

This appendix lists and describes the various fields and control devices in Oracle Directory Manager and the Oracle Internet Directory Self-Service Console.

Appendix D, "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 E, "The Access Control Directive Format"

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

Appendix F, "Addition of a Directory 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.

Appendix G, "Globalization Support in the Directory"

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

Appendix H, "Troubleshooting"

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

Related Documentation

For more information, see:

- Online help available through Oracle Directory Manager, the Oracle Delegated Administration Services and Oracle Enterprise Manager
- The Oracle Application Server and Oracle9*i* Database Server documentation sets, especially:
  - *Oracle Internet Directory Application Developer's Guide*
  - *Oracle Identity Management Concepts and Deployment Planning Guide*
  - *Oracle9i Database Administrator's Guide*
  - *Oracle9i Application Developer's Guide - Fundamentals*
  - *Oracle Application Server 10g Administrator's Guide*
  - *Oracle9i Net Services Administrator's Guide*
  - *Oracle9i Real Application Clusters Administration*
  - *Oracle9i Advanced Replication*
  - *Oracle Advanced Security Administrator's Guide*

Printed documentation is available for sale in the Oracle Store at

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

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

`http://otn.oracle.com/membership/`

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/documentation/`

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 available at: `http://www.ietf.org`, especially:
  - The LDAPEXT charter and LDAP drafts
  - The LDUP charter and drafts
  - RFC 2254, "The String Representation of LDAP Search Filters"
  - RFC 1823, "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 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.
`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`
		`----------`
		`/fs1/dbs/tbs_01.dbf`
		`/fs1/dbs/tbs_02.dbf`
		`.`
		`.`
		`.`
		`/fs1/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`

Convention	Meaning	Example

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

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

Conventions for Windows Operating Systems

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

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - *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`

Convention	Meaning	Example
`C:\>`	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the *command prompt* in this manual.	`C:\oracle\oradata>`
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	`C:\>exp 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`

Convention	Meaning	Example
ORACLE_HOME and *ORACLE_BASE*	In releases prior to Oracle8*i* release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory. For Windows NT, the default location was `C:\orant`. This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle`. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is `C:\oracle\ora`*nn*, where *nn* is the latest release number. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. Refer to *Oracle9i Database Platform Guide 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 10*g* (9.0.4)

- **Integration with the Microsoft Windows environment**—You can integrate the Oracle Application Server infrastructure with the Microsoft Windows Operating System—including Microsoft Active Directory and Microsoft Windows NT 4.0. This integration is achieved by using the Active Directory Connector in the Oracle Directory Integration and Provisioning platform and plug-ins.

  See Also: Chapter 43, "Integration with the Microsoft Windows Environment"

- **External authentication support**—You can store user security credentials in a repository other than Oracle Internet Directory—for example, a database or another LDAP directory such as Microsoft Active Directory or SunONE Directory Server. You can then use these credentials for user authentication.

  See Also:

  - Chapter 47, "Setting Up the Customized External Authentication Plug-in"
  - "Choose Where to Store Passwords" on page 41-6

- **Dynamic groups**—You can create and use dynamic groups whose membership, rather than being maintained in a list, is computed on the fly, based on assertions that you specify.

  See Also: Chapter 9, "Dynamic and Static Groups in Oracle Internet Directory"

- **Query optimization**—In searches, some attributes have very different response times depending on their values. You can uniform the response times of search operations for such attributes to enhance performance.

  See Also: "Optimizing Searches" on page 21-12

- **Garbage collection framework**—A garbage collector is a background database process that removes obsolete data from the directory. The Oracle Internet Directory garbage collection framework provides a default set of garbage collectors, and enables you to modify them.

  See Also: Chapter 22, "Garbage Collection in Oracle Internet Directory"

- **Simple Authentication Security Layer (SASL) support**—Oracle Internet Directory supports the use of SASL, a method for adding authentication support to connection-based protocols. To use it, a protocol includes a command for identifying and authenticating a user to a server and for optionally negotiating protection of subsequent protocol interactions. If its use is negotiated, a security layer is inserted between the protocol and the connection.

  See Also: "Authentication in Oracle Internet Directory" on page 12-4

- **Logging enhancements**—This release of Oracle Internet Directory provides the following enhancements to logging and tracing:
  - Object-based tracing for operations associated with thread and connection identifiers. This facilitates non-interleaved and coherent logging for each LDAP operation in a multithreaded environment.
  - Selective tracing for chosen operations by using the operation dimension
  - Structured, meaningful trace messages with additional information including thread identifier and criticality

  See Also: Chapter 10, "Logging, Auditing, and Monitoring the Directory"

- **OID Migration Tool (ldifmigrator) enhancements**—You can use this tool to reconcile data with that in an existing directory, and to directly load data into Oracle Internet Directory.

See Also:

- "Migrating User Data from Application-Specific Repositories" on page 23-5
- "The OID Migration Tool (ldifmigrator) Syntax" on page A-135

- **Client side referral caching**—This new feature enables clients to cache referral information and use it to speed up referral processing.

  See Also:

  - "Client-Side Referral Caching" on page 7-19
  - The material on the `ldap_set_option` and the `ldap_get_option` in *Oracle Internet Directory Application Developer's Guide*

- **Fan-out and partial replication support**—Oracle Internet Directory now supports:
  - Partial replication—that is, propagation of one or more naming contexts, rather than the entire DIT, to another node
  - Fan-out replication, in which a consumer, having received changes from a supplier, can then replicate those changes to one or more other consumers. Fan-out replication can be either full or partial.

  See Also:

  - Chapter 24, "Directory Replication Concepts"
  - Chapter 25, "Oracle Directory Replication Administration"

- **Password policy enhancements**—New password policy capabilities in Oracle Internet Directory include:
  - Password history
  - Unlocking of accounts
  - Forced password change upon first login

- Self-resetting of password in case of account lockout or forgotten passwords
- IP-based account lockout
- Password policy enablement or disablement by using a single attribute in the password policy entry

See Also: Chapter 15, "Password Policies in Oracle Internet Directory"

- **Security credential storage enhancements**—New security credential storage capabilities in Oracle Internet Directory include:
  - Generation of O3logon verifier for enterprise users
  - Generation of a default set of verifiers for application bootstrapping
  - Generation of SASL/MD5 verifiers for directory authentication

  See Also: Chapter 16, "Directory Storage of Password Verifiers"

- **Replication Environment Management Tool**—This tool ensures that Oracle9*i* Advanced Replication is properly configured for directory replication. In the event of a directory replication failure, this tool looks for common problems and seeks to rectify them. If it cannot solve the problem, then it gives you a report of the nature of the problem and points you to a possible solution.

  See Also: "The Replication Environment Management Tool" on page A-62

- **Server discovery by using DNS**—This feature enables the location of a directory server in a distributed environment to be discovered dynamically by using the domain name system (DNS). Rather than storing server location information statically in an `ldap.ora` file on the client, that information is stored and managed in a central domain name server. The client, at request processing time, retrieves this information from the domain name server.

  See Also: "The Replication Environment Management Tool" on page A-62

- **Bulkload tool enhancements**—You can now use bulkload to add a large volume of entries to a non-empty directory. For example, you can add one million entries to a directory that has one million entries already. You can also incrementally add a medium-size number of entries to a large directory. For example, you can add 50,000 entries at a time to a directory that has five million entries already.

  See Also: "bulkload Syntax" on page A-45

- **Rack-mounted directory server configuration support**—This configuration provides high availability of a directory server by running multiple directory server instances on different hardware nodes. The directory servers are connected to the same underlying data store, which is an Oracle9*i* Database Server.

  See Also: Chapter 27, "Rack-Mounted Directory Server Configurations"

- **Two-way provisioning between** Oracle Internet Directory **and other application directories**—The Oracle Directory Provisioning Integration Service can send notification of provisioning events bidirectionally between Oracle Internet Directory and other applications.

  See Also: "How an Application Receives Provisioning Information from Oracle Internet Directory" on page 34-7

  "How Oracle Internet Directory Receives Provisioning Information from an Application" on page 34-8

- **Integration of provisioning data with the Oracle E-Business Suite**—You can synchronize user accounts and other user information from the Oracle E-Business Suite to Oracle Internet Directory by using the Oracle Directory Provisioning Integration Service.

See Also: Chapter 40, "Integration of Provisioning Data with the Oracle E-Business Suite"

- **Installation of** Oracle Internet Directory **on Oracle9*i* Real Application Clusters**—You can install Oracle Internet Directory on Oracle9*i* Real Application Clusters. When you do this, both the software and schema for Oracle Internet Directory are installed on the primary node, while only the software is installed on the secondary nodes.

  See Also: The installation documentation for this release of Oracle Internet Directory

- **Oracle Directory Manager enhancements**—Oracle Directory Manager now enables you to manage the following:
  - Attribute uniqueness
  - Plug-ins
  - Garbage collection
  - Change logs
  - Replication
  - Query optimization
  - Debug logging to a finer degree than previously
  - Enhancement of ACLs
- **Oracle Internet Directory Self-Service Console enhancements**—Oracle Internet Directory Self-Service Console, a graphical administrative tool built with Oracle Delegated Administration Services units, enables you to manage the following:
  - Realms
  - Services
  - Accounts
  - Password resetting

Oracle Internet Directory Self-Service Console also enables you to view your organization chart, and users to edit their own profiles.

> **See Also:** Chapter 31, "Oracle Internet Directory Self-Service Console"

- **Upgrade procedures**

  > **See Also:** *Oracle Application Server 10g Upgrading to 10g (9.0.4)* for information about upgrading from an earlier version of Oracle Internet Directory

About Oracle Internet Directory Release 9.2

This section describes an important new feature employing the capabilities of Oracle Internet Directory. It also explains changes in Oracle Internet Directory since Release 9.0.2.

- **User Migration Utility for bulk-migrating database users to** Oracle Internet Directory—This utility, released with Oracle Advanced Security Release 2 (9.2), enables you to migrate users from a local or external database to Oracle Internet Directory. Use it to store and centrally manage thousands of users in Oracle Internet Directory.

  > **See Also:** The chapter about migrating local or external users to enterprise users in *Oracle Advanced Security Administrator's Guide*

Note:

- Beginning with Oracle Internet Directory Release 9.2, the Oracle Delegated Administration Services and tools built on it are components of Oracle Application Server and not the Oracle9*i* Database Server. To ensure that you have the self-management tools for administering Web and Oracle Application Server applications, and that those tools are well-integrated with your middle-tier environment, Oracle Corporation recommends that you use the version of Oracle Internet Directory that is included with the Oracle Application Server. To develop and deploy tools based on the Oracle Delegated Administration Services, Oracle Corporation recommends that you use the Java and security infrastructure of Oracle Application Server.
- Oracle Internet Directory Release 9.2 does not include Enterprise Manager integration for performing system diagnostics on Oracle Internet Directory instances.

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 21-11

- **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 and Provisioning 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 keeping local repositories in synch. 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 32, "Oracle Directory Integration and Provisioning 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 12-8 for a conceptual discussion
- Chapter 15, "Password Policies in Oracle Internet Directory" 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 the RDN dlin would be unique directly under `ou=people,o=oracle`. 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.

  Attributes other than `dn` can be used as unique keys of applications synchronizing with Oracle Internet Directory. The ability of Oracle Internet Directory to enforce attribute uniqueness enables all applications to have their

own notions of "user," and to synchronize their user base with a user repository stored in an enterprise Oracle Internet Directory server.

See Also: Chapter 8, "Attribute Uniqueness in the Directory"

- **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 X.509 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 16, "Directory Storage of Password Verifiers"

- **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 from Oracle 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 12, "Directory Security Concepts"
  - "Managing Super Users, Guest Users, and Proxy Users" on page 5-11

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

  See Also: Chapter 19, "Deployment of Oracle Identity Management Realms"

- **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 Oracle Internet Directory Servers on page 10-17

- **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 realms

  – 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 Oracle Enterprise Manager GUI can now be done through Oracle Directory Manager 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 45, "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 5-14

- **Delegated Administration Service**

  The Oracle Delegated Administration Services is a set of individual, pre-defined services—called Oracle Delegated Administration Services 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.

  Administrators can now use the Oracle Delegated Administration Services 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, a new component of the Oracle Delegated Administration Services, enables you to flexibly administer applications, realms, and end users either from a central team or through decentralization and delegation. It provides:

  – A unified resource for directory administrators, directory service subscribers, and end users

  – A view of an authorized end user's personalized preferences and the ability to update their Oracle Application Server Single Sign-On password

  – An intuitive user interface for searching for people and other directory-based resource information within Oracle Internet Directory.

You can use the Oracle Internet Directory 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 31, "Oracle Internet Directory Self-Service Console"

- **Upgrade procedures**

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

New Features Introduced with Oracle Internet Directory Release 3.0.1

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

- **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 28, "Cold Failover Cluster Configuration"

- **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 29, "The Directory 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.

See Also: Chapter 28, "Cold Failover Cluster Configuration"

- **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: "Multiple installations of Oracle Internet Directory on one host" on page 18-6

- **Oracle Directory Integration and Provisioning Platform**

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

  See Also: Part VII: "Oracle Directory Integration and Provisioning Platform"

- **Password policy management**

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

  See Also:

  - "Password Policies in Oracle Internet Directory" for a conceptual discussion
  - Chapter 15, "Password Policies in Oracle Internet Directory"

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

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

- **UTF8 restriction removed**

  The Oracle directory server and database tools are no long restricted to run on a UTF8 database. However, there may be data loss during add, delete, modify, or modifydn operations if the character sets of the data contained in the client request and the directory server database repository are different and the client data cannot be mapped to the database character set. If the database underlying the Oracle directory server is neither AL32UTF8 nor UTF8, then be sure that all characters in the client character set are included in the database character set, with the same or different character codes.

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 7-8
- "Managing Entries with Attribute Options by Using Command-Line Tools" on page 7-12

- **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 in Multimaster Replication" on page 22-7 for a conceptual discussion
  - "Viewing and Modifying Directory Replication Server Configuration Parameters" on page 25-36

- **Enhanced support for these operational attributes: creatorsName, createTimestamp, modifiersName, and 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-42 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 23, "Migration of Data from Other 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: "OID Database Statistics Collection Tool (oidstats.sh) Syntax" on page A-133

- **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 12-8 for a conceptual discussion
- Chapter 15, "Password Policies in Oracle Internet Directory" 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 Command-Line Tools" on page 4-14 for a brief explanation of this tool
    - "About the Human Intervention Queue Manipulation Tool" on page 25-21
    - "About the OID Reconciliation Tool" on page 25-22

- **Replication node deletion**

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

See Also: "Deleting a Node from a Multimaster Replication Group" on page 25-18

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

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

  Note: This feature was replaced in Release 3.0.1 by the Oracle Directory Integration and Provisioning platform. See Chapter 32, "Oracle Directory Integration and Provisioning 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 3.0.1.

Part I

Getting Started

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

- Chapter 1, "Introduction to LDAP and Oracle Internet Directory"
- Chapter 2, "Directory Concepts and Architecture"
- Chapter 3, "Preliminary Tasks and Information"
- Chapter 4, "Directory Administration Tools"

1

Introduction to LDAP and Oracle Internet Directory

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?

A directory is a way in which complex information is organized, making it easy to find. Directories list resources—for example, people, books in a library, or merchandise in a department store—and give details about each one. They can be either offline—for example, a telephone book or a department store catalog—or online.

Online directories are used by enterprises with distributed computer systems 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 e-mail addresses
- An application, such as a message transport agent, locating a user's mail server
- A database application identifying role information for a user

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.

Table 1–1 Comparison of Online Directories and 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-enabled 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 the Lightweight Directory Access Protocol (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)**, 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.orgl`
- "Related Documentation" on page -lvii for an additional list of resources on LDAP
- Chapter 2, "Directory 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:

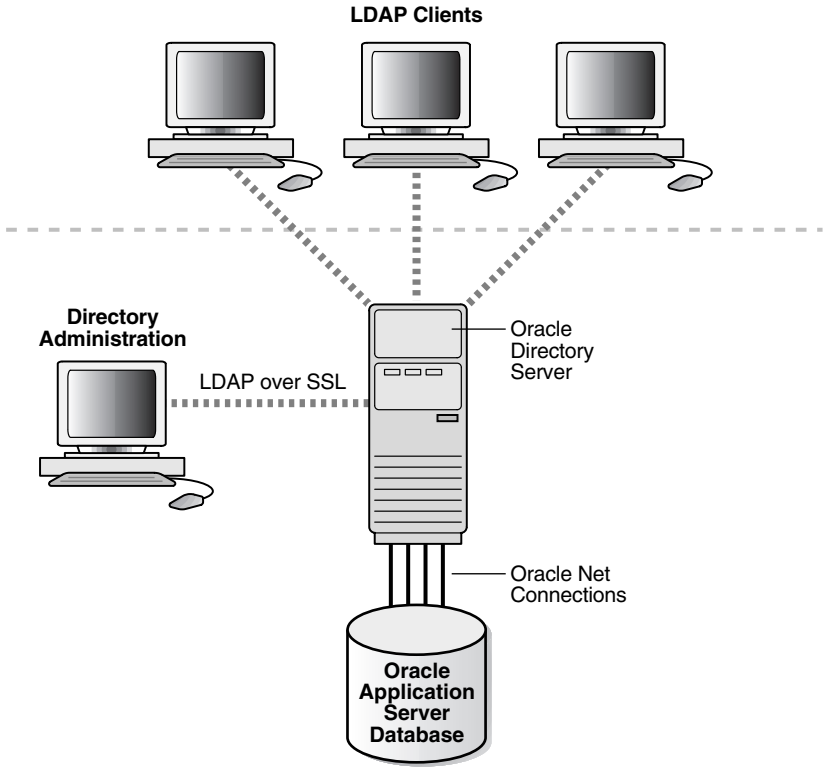
- Architecture of the Oracle Internet Directory
- Components of Oracle Internet Directory
- Advantages of Oracle Internet Directory

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 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
  - Directory server management tools within Oracle Enterprise Manager Application Server Control. These tools enable you to:
    * Monitor real-time events and statistics from a normal browser
    * Start the process of collecting such data into a new repository
- Oracle Internet Directory Software Developer's Kit

  See Also: *Oracle Internet Directory Application Developer's Guide* for information about the Oracle Internet Directory Software Developer's Kit

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 shared LDAP servers and database connection pooling enable 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 Advanced 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

Through the Oracle Directory Integration and Provisioning 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.

Oracle Identity Management

Oracle Internet Directory is a component of Oracle Identity Management, an integrated infrastructure that provides distributed security services for Oracle products and other enterprise applications. In addition to Oracle Internet Directory, the Oracle Identity Management infrastructure includes the following components and capabilities:

- Oracle Directory Integration and Provisioning platform: This component enables synchronization between Oracle Internet Directory and:
  - Other directories and user repositories
  - Automatic provisioning services for Oracle components and applications
  - Third-party applications
- Oracle Delegated Administration Services: This component provides trusted proxy-based administration of directory information by users and application administrators.
- Oracle Application Server Single Sign-On: This component provides single sign-on access to Oracle and third-party Web applications.

- Oracle Application Server Certificate Authority: This component generates and publishes X.509 V3 PKI certificates to support strong authentication methods.

To support enterprise application deployments, a single Oracle Identity Management infrastructure is typically deployed in the enterprise. It can include multiple server and component instances to provide high availability, information localization, and delegated component administration. Each additional application in the enterprise then leverages the shared infrastructure for identity management services. This deployment model has a number of advantages, including:

- Planning and implementing the identity management infrastructure is a one-time cost, rather than a necessary part of each enterprise application deployment. As a result, new applications such as portals, J2EE applications, and e-business applications can be rapidly deployed.
- Identities, while possibly administered in multiple places, are centrally managed and instantly available to all enterprise applications.
- A centralized security infrastructure makes it possible to realize user single sign-on across enterprise applications.
- A centralized identity management infrastructure provides a single point of integration between the enterprise Oracle environment and other identity management systems. This eliminates the need for multiple, custom, point-to-point integration solutions.

See Also:

- *Oracle Identity Management Concepts and Deployment Planning Guide* for information about planning, deploying and using the Oracle Identity Management infrastructure
- Chapter 19, "Deployment of Oracle Identity Management Realms" for a fuller discussion of the role of Oracle Internet Directory in relation to the Oracle Identity Management

How Oracle Components Use Oracle Internet Directory

Oracle components use Oracle Internet Directory for easier administration, tighter security, and simpler integration between multiple directories.

This section contains these topics:

- Easier and More Cost-Effective Administration of Applications
- Tighter Security Through Centralized Security Policy Administration

- Integration of Distributed Directories

Easier and More Cost-Effective Administration of Applications

OracleAS Portal enables self-service, integrated enterprise portals to store common user and group attributes in Oracle Internet Directory. The Oracle Portal administration tool also leverages the Oracle Delegated Administration Services for certain tasks.

Oracle Collaboration Suite uses Oracle Internet Directory for:

- Centralized management of information about users and groups
- Provisioning Oracle Collaboration Suite components—that is, notifying them whenever changes of interest are applied to data in Oracle Internet Directory
- Centralized integration for enterprises connecting other directories with any Oracle Collaboration Suite component

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.

Tighter Security Through Centralized Security Policy Administration

Oracle9*i* uses Oracle Internet Directory to store user names and passwords. It uses Oracle Internet Directory to store a password verifier along with the entry of each user.

Oracle Application Server 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.

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* Database Server and Oracle Application Server, 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.

Integration of Distributed Directories

The Oracle Directory Integration and Provisioning platform is a collection of interfaces and services for integrating multiple directories by using Oracle Internet Directory and several associated plug-ins and connectors.

The Oracle Directory Integration and Provisioning platform provides these benefits:

- All Oracle components are pre-certified to work with Oracle Internet Directory
- You can integrate the entire Oracle environment with third-party directories simply by integrating each third-party directory with Oracle Internet Directory.

This spares you the cumbersome task of integrating each application with each directory.

2

Directory 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 lvii for suggestions on further reading about LDAP-compliant directories

Entries

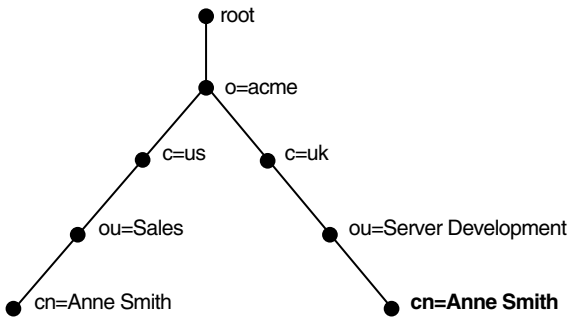
In an online directory, each collection of information about an object is called an **entry**. An entry can include, for example, information about an employee, a conference room, an e-commerce partner, or a shared network resource such as a printer.

Distinguished Names (DNs) and Directory Information Trees (DITs)

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 contained in 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 contained in 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 previous 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 concatenation of RDNs that reflects parent-child relationships in the DIT. Within the DN, RDNs are 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—for example, you could identify each employee with a unique number.

Entry Caching

To make operations on entries quick and efficient, Oracle Internet Directory uses entry caching. When you enable this feature, 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 enhances Oracle Internet Directory performance, and is especially useful in smaller and medium-sized enterprises.

Note: In Oracle Internet Directory 10*g* (9.0.4), you can use entry caching only in the case of a single server, single instance Oracle Internet Directory node.

See Also: Chapter 7, "Directory Entries Administration"

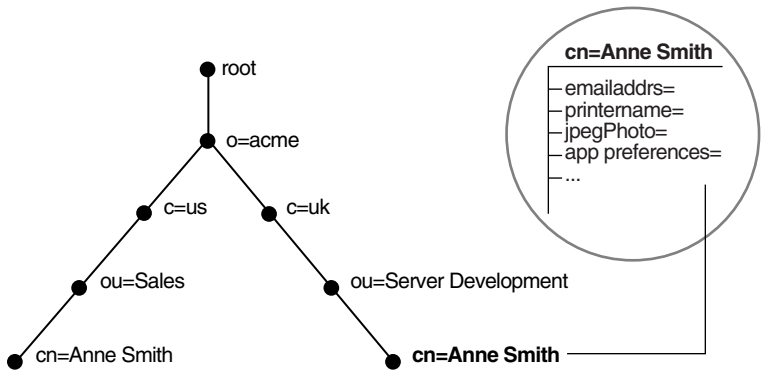
Attributes

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 Attributes

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

- Operational Attributes

  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.

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 5-9 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 as defined by RFC 2798 of the Internet Engineering Task Force (IETF).

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. This is stored in binary format.
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 B, "Oracle Internet Directory Schema Elements" for a list of several 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.

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

See Also: "LDAP Syntax" on page B-44

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 B-47

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 `givenName;lang-fr=Jean`, the base attribute is `givenName`; the French value for that base attribute is `givenName;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 7-8
- "Managing Entries with Attribute Options by Using Command-Line Tools" on page 7-12

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 and must have values, others are optional and can be empty.

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`.

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 a subclass 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 also inherit attributes that their superclasses have inherited.

> **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 definitions of the superclass.

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

Object Class Types

There are three types of object classes:

- Structural
- Auxiliary
- Abstract

Structural Object Classes

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

These object classes model real-world entities and their physical or logical attributes. Examples include people, printers, and database connections.

Structural object classes use structure rules to place restrictions on the kinds of objects 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 optional attributes that expand the existing list of attributes in an entry. Unlike structural object classes, they do not place restrictions on where an entry may be stored, and you can attach them to any entry regardless of that entry's location in the DIT.

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

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:

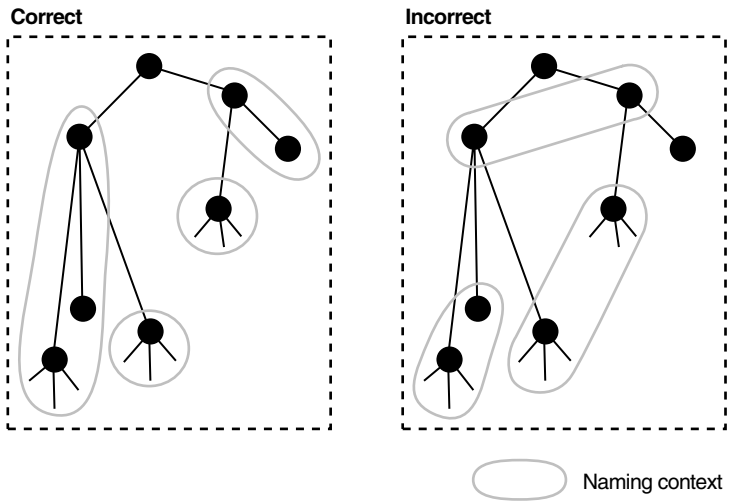
- "Globalization Support" on page 2-13 for more information on access control policies and ACLs
- "How to Extend the Number of Attributes Associated with Entries" on page 6-20 for a discussion of how to add additional content to entries

Naming Contexts

A **naming context** is a subtree that resides entirely on one server. It must be a complete subtree, 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 correct and incorrect naming contexts. Notice that the correct ones on the left are contiguous, and the incorrect ones on the right are not.

Figure 2–3 Correct and Incorrect Naming Contexts



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

> **See Also:** "Managing Naming Contexts" on page 5-10 for instructions on how to publish a naming context

Security

Oracle Internet Directory is a key element of the Oracle Identity Management Infrastructure. This enables you to deploy multiple Oracle components to work against a shared instance of Oracle Internet Directory and associated infrastructure

pieces. This sharing allows an enterprise to simplify security management across all applications.

In addition to the role it plays in the Oracle Identity Management infrastructure, Oracle Internet Directory provides many powerful features for protecting information.

These security features within Oracle Internet Directory itself include:

- Data integrity: Ensuring that data is not tampered with during transmission
- Data privacy: Ensuring that data is not inappropriately observed during transmission between Oracle Internet Directory and other components in the network.
- 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 that passwords are not easily discovered by others

In either an enterprise or hosted environment, you can use all these features to enforce a uniform security policy across multiple applications enabled for use with Oracle Internet Directory. 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 12, "Directory Security Concepts" for a fuller discussion of the security features of Oracle Internet Directory
- Chapter 19, "Deployment of Oracle Identity Management Realms" for information on Oracle Internet Directory as it relates to the Oracle Identity Management infrastructure
- Chapter 17, "Delegation of Privileges for an Oracle Technology Deployment" for a discussion of how to protect applications in a large enterprise and in hosted environments
- Chapter 36, "Security in the Oracle Directory Integration and Provisioning Platform" for a discussion of the unique aspects of security in an Oracle Directory Integration and Provisioning platform environment
- *Oracle Identity Management Concepts and Deployment Planning Guide* for a fuller discussion of the Oracle Identity Management infrastructure

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. (The Oracle character set name is AL32UTF8.) 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 means support for 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. An ASCII representation of a simplified Chinese directory entry definition 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 an ASCII representation of a simplified Chinese directory entry definition.

By default, 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)—accept only the UTF-8 character set. The Oracle character set name is AL32UTF8.

The Oracle directory server and database tools are no longer restricted to run on a UTF8 database. However, be sure that all characters in the client character set are included in the database character set (with same or different character codes) if the database underlying the Oracle Internet Directory server is not AL32UTF8 or 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 Directory Manager, a Java-based tool, internally uses **Unicode** (**UTF-16**—that is, fixed-width 16-bit Unicode). It can support internationalized character sets.

See Also:

- "Oracle Internet Directory Architecture" on page 2-14 for information on the main Oracle Internet Directory components
- Appendix G, "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

Oracle Internet Directory Architecture

This section contains these topics:

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

An Oracle Internet Directory Node

An Oracle Internet Directory node consists of one or more directory server instances connected to the same directory store. The directory store—that is, the repository of the directory data—is an Oracle9*i* Database Server.

Figure 2–4 on page 2-16 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:

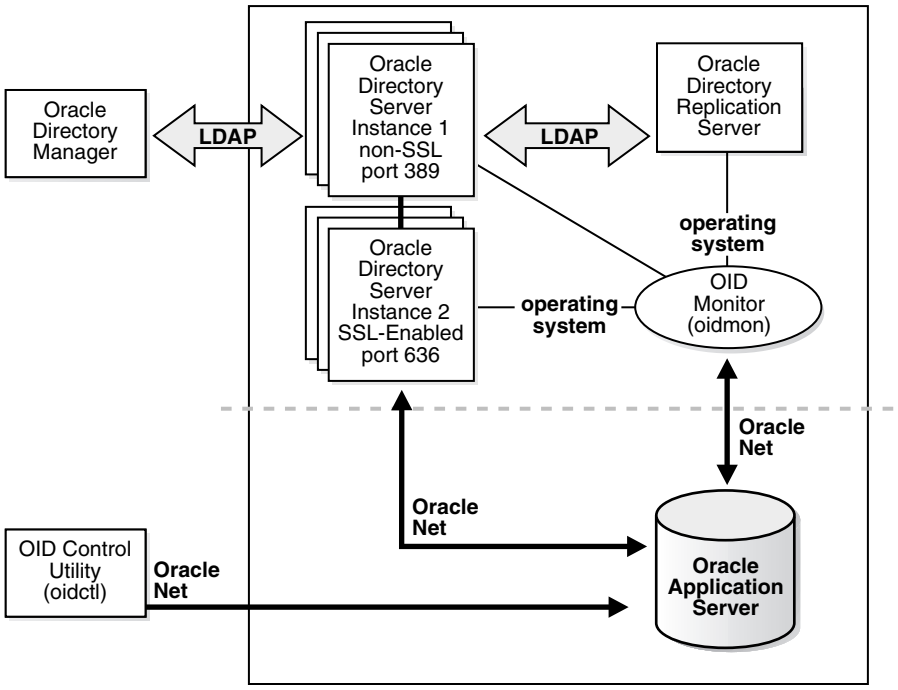
- The **OID Control Utility**
- The Oracle directory server instance 1 non-SSL port 389
- The Oracle directory server instance 2 SSL-enabled port 636
- The **OID Monitor**

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



As shown in Figure 2–4, an Oracle Internet Directory node includes the following major components:

Table 2–2 Components of an Oracle Internet Directory Node

Component	Description
Oracle directory server instance	Also called either an LDAP server instance or a directory server instance, it 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.
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 configure the replication server.

Table 2–2 (Cont.) Components of an Oracle Internet Directory Node

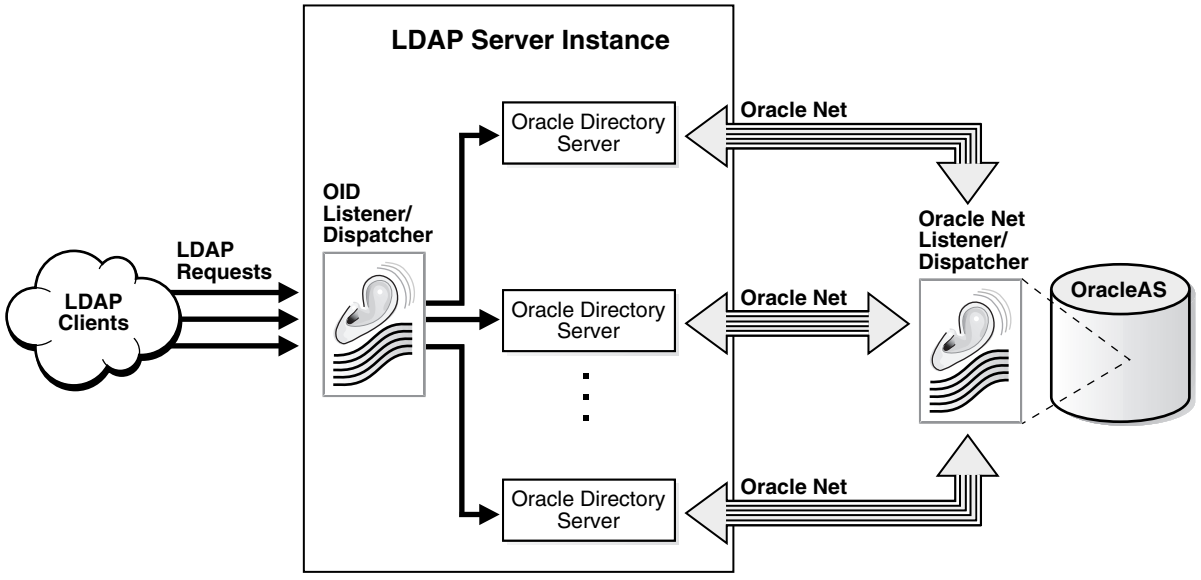
Component	Description
Oracle9*i* database	Stores the directory data. Oracle Corporation strongly recommends that you dedicate a database for use by the directory. The database can reside on the same node as the directory server instances.
OID Monitor (OIDMON)	Initiates, monitors, and terminates the LDAP 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



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.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. A server process handles an LDAP operation request and connects to the Oracle database instance to access the directory store. The directory server handles the client request by generating one server process for each operation.

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).

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.

Directory Metadata

Directory metadata is the information used by the directory server during run time for processing LDAP requests. It is stored in the underlying data repository. During startup, the directory server reads this information and stores it in a local metadata cache. It then uses this cache during its runtime to process incoming LDAP operation requests.

The directory server has the following types of metadata in its local metadata cache.

- Directory Schema

  The definitions of object classes, attributes, and matching rules supported by the directory server. The directory server uses this information during creation and modification of directory objects. A directory object is a collection of object classes and their associated attributes and matching rules.

- Access control policy point (ACP)

  A directory administrative domain for defining and controlling access to the information in that domain. The directory server uses ACPs when determining whether to allow a certain LDAP operation performed by a user.

- Root DSE entry

  The root DSE (DSA-Specific Entry) contains a number of attributes that store information about the directory server itself. For example, these attributes contain the following information items, to mention just a few:

  – Naming contexts DNs

  – Sub Schema Subentry DN

  – Superior references (referrals) DNs

  – Special entry DNs like Oracle Internet Directory configuration and registry containers

  – Special Entry DNs like change log and change status containers

  – DN of replications agreement container

- Privilege groups

  Groups that can be used in access control policies.

  The directory schema supports directory group objects through the standard `groupofuniquenames` and `groupofnames` object classes. These object classes hold information for such groups as distribution lists and mailing lists to mention just two.

  Oracle Internet Directory extends these standard group objects through an auxiliary object class called `orclprivilegegroup`. This object class, which supports privilege groups that can be used in access control policies, provides flexibility to grant or deny access to groups of users. The directory server uses this information during:

  – LDAP bind operations to find out the subscribed privileged groups for a given user

  – Access control policy evaluation if the policy has directives that grant or deny access to privileged groups

  Instructions on how to modify a group entry to associate it with or disassociate it from an object class can be found in either "Modifying Entries by Using Oracle Directory Manager" on page 7-7 or "Example: Modifying a User Entry by Using ldapmodify" on page 7-12.

- Catalog entry

  A special entry containing information about indexed attributes in the underlying database. The directory uses this information during directory search operations.

- Common entry

  A special entry containing information about hosted companies. A hosted company is an enterprise to which another enterprise provides services. The metadata in this entry includes the hosted company DN, user search base, nickname and other attributes, all of which are described in Chapter 19, "Deployment of Oracle Identity Management Realms".

- Plug-in entry

  A special entry containing information about the kind of operation that triggers a plug-in event, and the point in the operation when that plug-in is to be triggered. This information is described in Chapter 45, "Oracle Internet Directory Plug-in Framework".

- Password verifier entry

A special entry containing information about the encryption and verifier attribute types. This information is described in Chapter 16, "Directory Storage of Password Verifiers".

- Password policy entry

  A special entry containing information about the policies enforced by the directory server for the user password credentials. The directory server uses this information during runtime to enforce the password policies.

Configuration Set Entries

The configuration parameters for each Oracle directory server instance are stored in an entry called a configuration set entry, or configset. 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 configuration set entries 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 with parameters to meet your specific needs.

You can view, add, and modify configuration set 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 Schema Elements" on page B-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 be physically distributed onto several servers. This distribution 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, one or more unique, non-overlapping naming contexts are stored on each directory server. In a distributed directory, some information may be partitioned and some may be replicated.

This section contains these topics:

- Directory Replication
- Directory Partitioning

Directory Replication

Replication is the process of copying and maintaining the same naming contexts on multiple directory servers. It improves performance by providing more servers to handle queries, and reliability by eliminating risks associated with a single point of failure.

Replication can be either full or partial.

Full replication involves propagating the entire DIT to another node.

Partial replication involves propagating one or more subtrees, rather than the entire DIT, to another node.

The directory servers that participate in the replication of a given naming context form what is called a directory replication group (DRG). The relationship among the directory servers in a DRG is represented on each node by a special directory entry called a replication agreement.

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

A directory replication group can be either single-master, multimaster, or fan-out.

A single-master replication group has only one supplier replicating changes to one or more consumers. Only the supplier can be updated, and consumers are read-only.

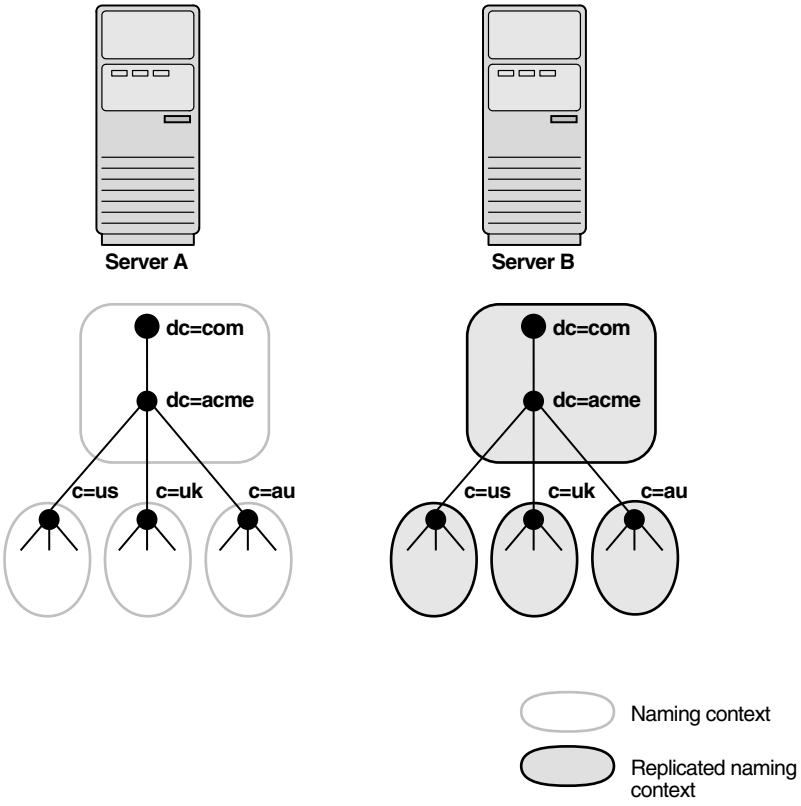
Multimaster replication, also called peer-to-peer or *n*-way replication, enables multiple sites, acting as equals, to manage groups of replicated data. In a multimaster replication environment, each node is both a supplier and a consumer node, and the entire directory is replicated on each node.

A fan-out replication group, also called a point-to-point replication group, has a supplier replicating directly to a consumer. That consumer can then replicate to one or more other consumers. The replication can be either full or partial.

In a directory replication group, the protocol for transferring data between nodes can be based on either Oracle9*i* Advanced Replication or LDAP.

Figure 2–6 shows a replicated directory.

Figure 2–6 A Replicated Directory



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

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**. It can use standard LDAP as a transport for transmitting these change logs between Oracle Internet Directory replicas.

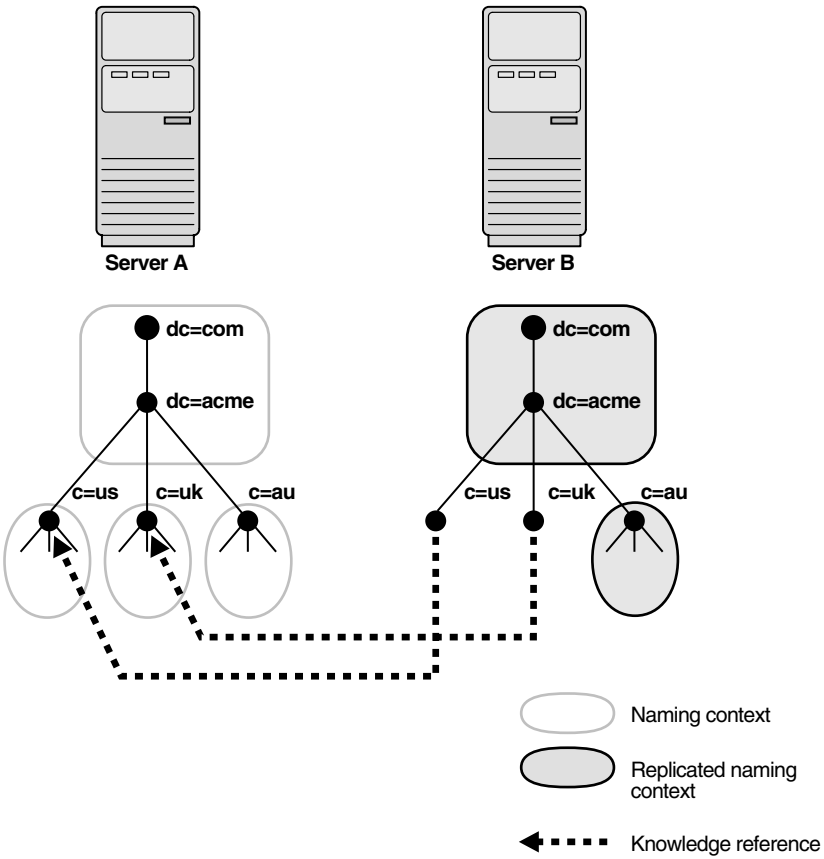
See Also: Chapter 24, "Directory Replication Concepts" for a more detailed discussion of replication, including: Oracle9*i* Advanced Replication architecture, LDAP-based replication, change log purging, conflict resolution, and the replication process

Directory 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 on page 2-26, four naming contexts reside on Server A:

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

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

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

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.**

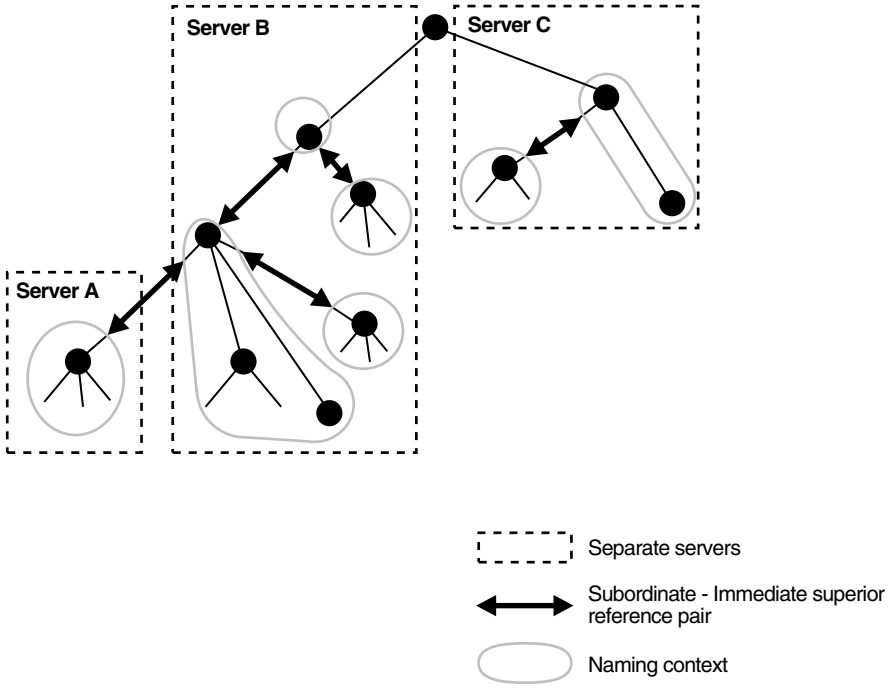
Knowledge References and Referrals

A knowledge reference provides the names and addresses of the various naming contexts held in another partition. For example, in Figure 2–7 on page 2-26, Server B uses knowledge references to point to the `c=us` and `c=uk` naming contexts on Server A. When Server B is asked for information residing on Server A, it sends back 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 on page 2-28, 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.

There are two kinds of referrals:

- Smart referrals

  These are 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 referrals

  These are 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 detailed information 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 7-17

Oracle Delegated Administration Services and the Oracle Internet Directory Self-Service Console

Oracle Delegated Administration Services is a set of pre-defined, Web-based units for performing directory operations on behalf of a user. This set of services frees directory administrators from the routine tasks of directory management by enabling them to delegate specific functions to other administrators and to end users. It provides most of the functionality that directory-enabled applications require, such as creating a user entry, creating a group entry, searching for entries, changing user passwords, and other employee-specific data.

You can use Oracle Delegated Administration Services to develop your own tools for administering application data in the directory. Or you can use the Oracle Internet Directory Self-Service Console, a tool based on Oracle Delegated Administration Services that comes ready-to-use with Oracle Internet Directory. This console is used by several Oracle components to provide delegated administration.

See Also:

- Chapter 30, "Oracle Delegated Administration Services"
- Chapter 31, "Oracle Internet Directory Self-Service Console"

The Oracle Directory Integration and Provisioning Platform

The Oracle Directory Integration and Provisioning platform enables an enterprise to integrate its applications and other directories with Oracle Internet Directory. It provides all the interfaces and infrastructure necessary to keep the data in Oracle Internet Directory consistent with that in enterprise applications and connected directories. It also makes it easier for third-party vendors and developers to develop and deploy their own connectivity agents.

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 OracleAS Portal) that need to be notified whenever changes are applied to Oracle Internet Directory.

Based on the nature of integration, the Oracle Directory Integration and Provisioning 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 to reflect changes made to a entries of interest, such as users and groups

  See Also: "Oracle Directory Integration and Provisioning Platform"

Oracle Internet Directory and Identity Management

Identity management is the process by which the complete security life cycle for network entities is managed in an organization. Because Oracle Internet Directory is a key element of the Oracle Identity Management infrastructure, it enables you to simplify security management across all applications. To do this, you deploy multiple Oracle components against a shared instance of Oracle Internet Directory and of other Oracle Identity Management components. This requires careful planning to match the Oracle Internet Directory deployment with the security needs of your enterprise.

This section contains these topics:

- About Identity Management
- About the Oracle Identity Management Infrastructure
- Identity Management Realms

About Identity Management

Identity management most commonly refers to the management of an organization's application users. Steps in their security life cycle include account creation, suspension, privilege modification, and account deletion. The managed entities may also include devices, processes, applications, or anything else that needs to interact in a networked environment. They may also include users outside of the organization, for example customers, trading partners, or Web services.

Identity management is important to IT deployments because it can reduce administrative costs while at the same time improving security.

The Oracle Identity Management infrastructure enables deployments to manage centrally and securely all enterprise identities and their access to various applications in the enterprise. Identity management comprises these tasks:

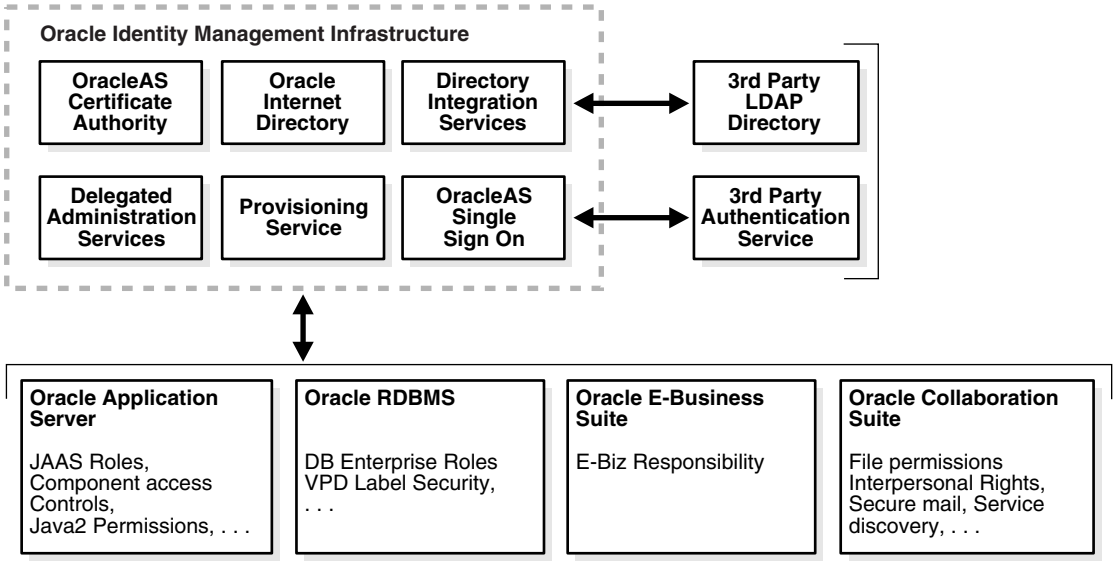
- Creating enterprise identities and managing shared properties of these identities through a single enterprise-wide console

- Creating groups of enterprise identities
- Provisioning these identities in various services available in the enterprise. This includes:
  - Account creation
  - Account suspension
  - Account deletion
- Managing policies associated with these identities. These include:
  - Authorization policies
  - Authentication Policies
  - Privileges delegated to existing identities

About the Oracle Identity Management Infrastructure

Oracle Identity Management is an integrated infrastructure that Oracle products rely on for distributed security. It is part of the infrastructure of the Oracle Application Server and for other Oracle products as well. Figure 2–9 on page 2-33 illustrates the components of the Oracle Identity Management infrastructure and how various Oracle and third-party products rely on it.

Figure 2–9 Oracle Identity Management Infrastructure and Other Components



As shown in Figure 2–9, the Oracle Identity Management infrastructure includes the following components and capabilities:

- Oracle Internet Directory, a scalable, robust LDAP Version 3-compliant directory service implemented on the Oracle9*i* Database Server.
- Oracle Directory Integration and Provisioning platform, which permits synchronization between Oracle Internet Directory and other directories and user repositories and automatic provisioning services for Oracle components and applications and, through standard interfaces, third-party applications.
- Oracle Delegated Administration Services, which provides trusted proxy-based administration of directory information by users and application administrators.
- Oracle Application Server Single Sign-On, which provides single sign-on access to Oracle and third party web applications.
- Oracle Application Server Certificate Authority, which generates and publishes X.509 V3 PKI certificates to support strong authentication methods.

While Oracle Identity Management is designed to provide an enterprise infrastructure for Oracle products, it can also serve as a general-purpose identity management solution for user-written and third-party enterprise applications. It

provides a robust and scalable enterprise-wide identity management platform for third-party applications, hardware, and network operating systems. Custom applications can leverage Oracle Identity Management through a set of documented and supported services and APIs, for example:

- Oracle Internet Directory provides LDAP APIs for C, Java, and PL/SQL, and is compatible with other LDAP SDKs.
- Oracle Delegated Administration Services provide a core self-service console that may be customized to support third-party applications. In addition, it provides a number of services for building customized administration interfaces that manipulate directory data.
- The Oracle Directory Synchronization Service facilitates the development and deployment of custom solutions for synchronizing Oracle Internet Directory with third-party directories and other user repositories.
- The Oracle Directory Provisioning Integration Service enables you to provision third-party applications and integrate the Oracle environment with other provisioning systems.
- Oracle Application Server Single Sign-On provides APIs for developing and deploying partner applications that share a single sign-on session with other Oracle Web applications.
- JAZN, Oracle's implementation of the JAAS standard, enables applications developed for the Web using Oracle's J2EE environment to leverage the Oracle Identity Management infrastructure for authentication and authorization.

In addition, Oracle works with third-party application vendors to ensure that their applications can leverage Oracle Identity Management out of the box.

> **See Also:** *Oracle Identity Management Concepts and Deployment Planning Guide* for more information about the Oracle Identity Management infrastructure

Identity Management Realms

An identity management realm defines an enterprise scope over which certain identity management policies are defined and enforced by the deployment. It comprises:

- A well-scoped collection of enterprise identities—for example, all employees in the US domain

- A collection of identity management policies associated with these identities. An example of an identity management policy would be to require that all user passwords have at least one alphanumeric character.
- A collection of groups—that is, aggregations of identities—that simplifies the setting of the identity management policies

You can define multiple identity management realms within the same Oracle Identity Management infrastructure. This enables you to isolate user populations and enforce a different identity management policy—for example, password policy, naming policy, self-modification policy—in each realm.

Each identity management realm is uniquely named to distinguish it from other realms. It also has a realm-specific administrator with complete administrative control over the realm.

Default Identity Management Realm

For all Oracle components to function, an identity management realm is required. One particular realm, created during installation of Oracle Internet Directory, is called the default identity management realm. It is where Oracle components expect to find users, groups, and associated policies whenever the name of a realm is not specified.

There can be only one default identity management realm in the directory. If a deployment requires multiple identity management realms, then one of them must be chosen as the default.

Identity Management Policies

The Oracle Identity Management infrastructure supports a flexible set of management policies which comprise:

- Directory structure and naming policies that enable you to:
  - Customize the directory structure in Oracle Internet Directory for your deployment
  - Specify where various identities are to be located and how they are uniquely identified
- Authentication policies that enable you to specify authentication methods and protocols supported by the Oracle Identity Management infrastructure
- Identity management authorizations that enable you to control access to certain privileged services and delegate administration wherever necessary

Note: In Oracle Internet Directory Release 9.0.2, the equivalent term for "identity management realm" was “subscriber”.

Resource Information

To fulfill the requests of users, some Oracle components gather data from various repositories and services. To gather the data, these components require the following information:

- Information specifying the type of resource from which the data is to be gathered. The type of resource could be, for example, an Oracle Database. This is called resource type information.
- Information for connecting and authenticating users to the resources. This is called resource access information.

This section contains these topics:

Resource Type Information

Information about the resources that an application uses to service a user request is called resource type information. A resource type can be, for example, an Oracle9*i* Database Server or a Java Database Connectivity Pluggable Data Source. Resource type information includes such items as the class used to authenticate a user, the user identifier, and the password.

You specify resource type information by using the Oracle Internet Directory Self-Service Console.

Resource Access Information

Information for connecting and authenticating users to the databases is called resource access information. It is stored in an entry called a resource access descriptor (RAD) from which it can be retrieved and shared by various Oracle components.

For example, to service the request of a user for a sales report, Oracle Application Server Reports Services queries multiple databases. When it does this, it does the following:

1. Retrieves the necessary connect information from the RAD
2. Uses that information to connect to those databases and to authenticate the user requesting the data

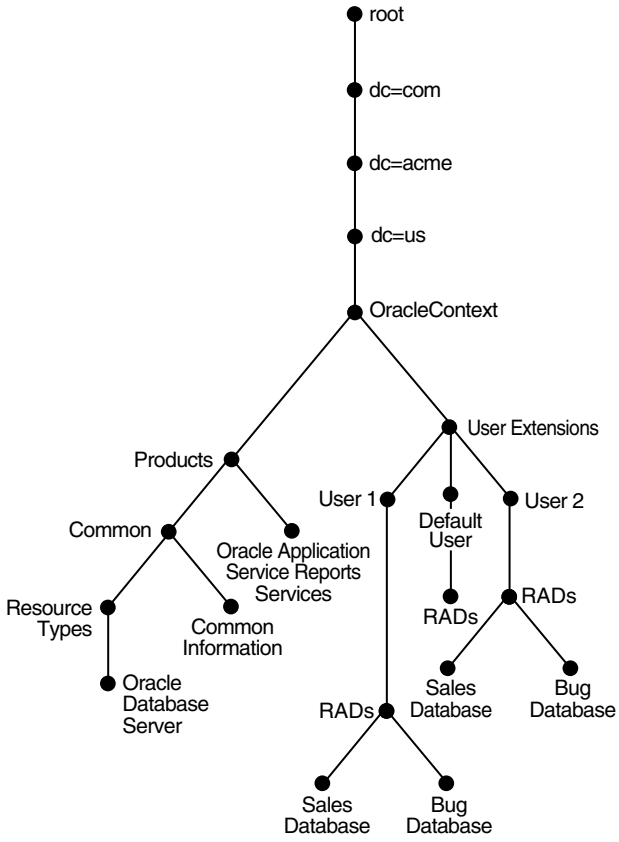
Once it has done this, it compiles the report.

You specify resource access information by using the Oracle Internet Directory Self-Service Console. You can specify resource access information for each individual user or commonly for all users. In the latter case, all users connecting to a given application use, by default, the same information to connect to the necessary databases. Oracle Corporation recommends defining default resource access information whenever an application has its own integrated account management—for example, where each user is defined within the application itself by means of a unique single sign-on user name.

Location of Resource Information in the DIT

Figure 2–10 shows where resource information is located in the DIT.

Figure 2–10 Placement of Resource Access and Resource Type Information in the DIT



As Figure 2–10 shows, the resource access and resource type information is stored in the Oracle Context.

Resource access information for each user is stored in the `cn=User Extensions` node in the Oracle Context. In this example, the `cn=User Extensions` node contains resource access information for both the default user and for specific users. In the latter cases, the resource access information includes that needed for accessing both the Sales and the Bug databases.

Resource access information for each application is stored in the object identified by the application name—in this example, `cn=Oracle Application Server Reports Services, cn=Products,cn=Oracle Context,dc=us,dc=acme,dc=com`. This is the user information specific to that product.

Resource type information is stored in the container `cn=resource types, cn=common,cn=products,cn=Oracle Context`.

See Also:

- "Managing Resource Access Information" on page 31-9 for instructions for an end user to specify resource access information
- Creating User Entries by Using the Oracle Internet Directory Self-Service Console on page 31-17 for instructions for an administrator to specify resource access information when creating user entries
- "Configuring Default Resource Access Information" on page 31-26 for instructions for an administrator to define commonly used resources that all users automatically inherit
- "Configuring Resource Type Information" on page 31-25 for instructions for an administrator to specify resource types
- "Plug-in Schema Elements" on page B-32
- *Oracle Application Server Reports Services Publishing Reports to the Web*

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.

> **See Also:** "The OID Monitor (oidmon) Syntax" on page A-4 for instructions on starting and stopping the OID Monitor

Task 2: Start a Server Instance

Once the OID Monitor is running, start a server instance by using either the Oracle Enterprise Manager Application Server Control or the OID Control Utility.

> **See Also:**
>
> - "Starting a New Directory Server Instance by Using Oracle Enterprise Manager Application Server Control" on page 10-23
> - "Starting, Stopping, Restarting, and Monitoring Oracle Internet Directory Servers" on page A-4
> - "Troubleshooting Directory Server Instance Startup" on page A-9

Note: You can run multiple instances if the directory server is on the same computer. For example, you can run one instance in SSL mode and another in non-SSL mode.

Task 3: Reset the Default Security Configuration

To meet the needs of your environment, you must customize the default security configuration. Table 3–1 lists and describes the tasks you must perform to do this.

Table 3–1 Tasks to Reset the Default Security Configuration

Task Area	Description
Protect the `subSchemaSubEntry` subentry and its children	Information about the directory is contained in the subentry `subSchemaSubEntry` and its children. Oracle Corporation recommends that you control access to these objects.

Table 3–1 (Cont.) Tasks to Reset the Default Security Configuration

Task Area	Description
Establish access to entries	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 directory entries
Modify default access policies	Oracle Internet Directory is installed with a default security configuration described in Chapter 17, "Delegation of Privileges for an Oracle Technology Deployment". Before you begin using the directory, you can modify this default configuration to meet the needs of your environment and ensure that each user has the appropriate authorization.
Modify the default password policy	Password polices are sets of rules that govern how passwords are used. Oracle Internet Directory is installed with a default password policy that you can modify to meet the needs of your environment.

See Also:

- Chapter 2, "Directory Concepts and Architecture" for an introduction to security features of Oracle Internet Directory, and to the default DIT for Oracle components using Oracle Internet Directory
- Chapter 14, "Directory Access Control" for a detailed explanation of access control options and instructions for setting up security
- Chapter 19, "Deployment of Oracle Identity Management Realms" for a detailed explanation of the Oracle Context schema
- "Default Password Policy" on page 15-2 for an explanation of the default password policy

Caution: Be careful when modifying the default ACLs in any Oracle Context. Doing so can disable the security of Oracle components in your environment. See component-specific documentation for details on whether you can safely modify the default ACLs in an Oracle Context.

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 Oracle Internet Directory is the same as password specified for the Oracle Application Server administrator (the ias_admin) during installation. You can change this password by using the OID Database Password Utility.

See Also: "OID Database Password Utility (oidpasswd) Syntax" on page A-131 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 shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

See Also: "OID Database Statistics Collection Tool (oidstats.sh) Syntax" on page A-133

Log File Locations

The Oracle Internet Directory components output their log and trace information to log files in the *ORACLE_HOME* environment. Table 3–2 lists each component and the location of its corresponding log file.

Table 3–2 Log File Locations

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`
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
- Routine Administration at a Glance

Directory administration is also aided by the Oracle Delegated Administration Services, a set of pre-defined, Web-based units for performing directory operations on behalf of a user. It frees directory administrators from the more routine directory management tasks by enabling them to delegate specific functions to other administrators and to end users. You can use it, for example, to enable end users to modify their personal profile information (including Oracle Application Server Single Sign-On passwords) without requiring the intervention of an administrator.

One tool, created by using Oracle Delegated Administration Services, is the Oracle Internet Directory Self-Service Console. This ready-to-use application provides a single graphical interface for delegated administrators and end users to manage data in the directory.

See Also:

- Chapter 30, "Oracle Delegated Administration Services"
- Chapter 31, "Oracle Internet Directory Self-Service Console"

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 by Using Oracle Directory Manager
- Navigating Oracle Directory Manager
- Connecting to Additional Directory Servers by Using Oracle Directory Manager
- Disconnecting from a Directory Server by Using Oracle Directory Manager
- 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 2-14 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* > Integrated Management > Oracle Directory Manager
UNIX	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 by Using Oracle Directory Manager

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.

   Choose **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.

Table 4–1 Fields in the Credentials Tab Page

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 entry by using the correct format, for example, `cn=Susie Brown,ou=HR,o=acme,c=us`
Password	If you are logging in as the super user and you specified a password for the super user during installation, in the **Password** field, 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** filed 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 5-11 for instructions on how to change the password

Table 4–1 (Cont.) Fields in the Credentials Tab Page

Field	Description
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** list. The Select Directory Servers dialog box displays a list of available servers. 2. Select a server. 3. Choose **OK**. To add a directory server to the list: 1. In the Select Directory Servers dialog box, choose **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. Choose **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. Choose **Edit**. The Directory Server Connection dialog box appears. 3. Modify the **Server** and **Port** fields, then choose **OK**. The modifications for that server appear in the list in the Select Directory Server dialog box.
Port	The default port (389) appears in this field. If there is more than one directory server instance on the same host, then each directory server instance has a different port, and, when you select the directory server instance, that port number appears in this field. To change this port number: 1. Choose the button to the right of the **Server** field. 2. In the Select Directory Server dialog box, select the directory server. 3. Choose **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 choose **OK**.

Table 4–1 (Cont.) Fields in the Credentials Tab Page

Field	Description
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 fails. ■ You have already created a wallet containing a certificate and a list of trusted certificates.

See Also:

- Chapter 13, "Secure Sockets Layer (SSL) and the Directory" for instructions on enabling SSL
- "Entries" on page 2-2 for instructions on formatting distinguished names
- "Configuring SSL Parameters" on page 13-3 for information about changing ports and their impact on security
- *Oracle Advanced Security Administrator's Guide* for instructions on creating a wallet by using Oracle Wallet Manager when using SSL

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

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

Table 4–2 Fields in the SSL Tab Page

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. Choose **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 choose 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

Table 4–3 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.

Table 4–3 Oracle Directory Manager Menu Bar

Menu	Menu Items
File	**Create**—Adds an object
	Create Like—Adds a new object by using the object selected in the navigator pane as a template
	Connect—Connects to a directory server selected in the navigator pane
	Disconnect—Disconnects from a directory server selected in the navigator pane
	Exit—Exits Oracle Directory Manager

Table 4–3 (Cont.) Oracle Directory Manager Menu Bar

Menu	Menu Items
Edit	**Edit**—Modifies an object **Remove**—Removes a selected object **Find Object Classes** or **Find 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 menu item searches for an object class. If you navigate to Oracle Internet Directory > ***directory server instance*** > **Server Management** > **Attributes**, then it searches for attributes.
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.
Operation	**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 **Configure Search Filter**—Narrows the range of entries the navigator pane displays according to whatever filter you specify **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**—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

Table 4–3 (Cont.) Oracle Directory Manager Menu Bar

Menu	Menu Items
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 4–4 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

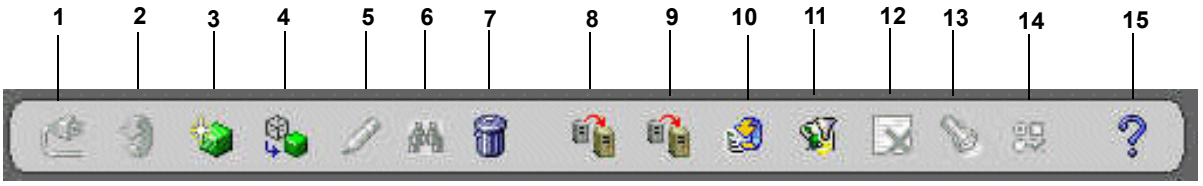


Table 4–4 Oracle Directory Manager Toolbar

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, then it searches for attributes.

Table 4–4 (Cont.) Oracle Directory Manager Toolbar

Button	Purpose
7	**Delete**—Removes an object
8	**Add Object Classes**—Adds an object class to an existing entry
9	**Refresh Entry**—Updates data for entries stored in memory to reflect changes in the database
10	**Refresh Subtree Entries**—Updates the children of entries stored in memory to reflect changes in the database
11	**Configure Search Filter**—Narrows the range of entries the navigator pane displays according to whatever filter you specify
12	**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.
13	**Search**—Enables you to configure ACP searches
14	**User Preferences**—Enables you to configure the display of ACPs in the navigator pane, as well as entries in a search operation
15	**Help**—Displays the Help system

Connecting to Additional Directory Servers by Using Oracle Directory Manager

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, choose **New**.
3. Follow the login procedures described in "Connecting to a Directory Server by Using Oracle Directory Manager" on page 4-3.

Disconnecting from a Directory Server by Using Oracle Directory Manager

To disconnect from a directory server by using Oracle Directory Manager, from the **File** menu choose **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.

Configuring the Display and Duration of Searches in Oracle Directory Manager

You can specify the maximum number of entries to be displayed in Oracle Directory Manager as the result of searches and the duration of searches. You can make these configurations in either Oracle Directory Manager or the directory server or both.

If you make the configuration in both Oracle Directory Manager and the directory server, and the configuration in Oracle Directory Manager does not match the one in the directory server, then Oracle Internet Directory resolves the conflict as follows:

- If the value you set in Oracle Directory Manager is greater than that in the directory server, then the configuration of the server prevails. For example, if you set Oracle Directory Manager to search for 2 minutes, and the directory server for 3 minutes, then the actual search duration will be 3 minutes.
- If the value you set in Oracle Directory Manager is less than that in the directory server, then the configuration of Oracle Directory Manager prevails. For example, if you set Oracle Directory Manager to search for 2 minutes, and the server for 3 minutes, then the actual search duration is 2 minutes.

To configure the display and duration of searches in Oracle Directory Manager:

1. In the navigator pane, expand Oracle Internet Directory **Servers**, and select the server you want to configure.
2. From the toolbar, select **User Preferences**. The User Preferences dialog box appears.
3. In the **Configure Entry Management** tab page, in the **Maximum number of one-level subtree entries** field, enter the maximum number of entries to be returned by a search.
4. In the **Search Time Limit** field, enter the maximum number of seconds for a search to be completed. The default is 3600.
5. Choose **OK**.

To configure the display and duration of searches in an Oracle directory server:

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. In the **Server Operation Time Limit** field, enter the maximum number of seconds for a search to be completed. The default is 3600.
4. Choose **Apply**.

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) and starting and stopping server instances. To perform tasks that you cannot perform with Oracle Directory Manager, use the appropriate LDAP command-line tool.

See Also:

- "Using Command-Line Tools" on page 4-14
- Appendix A, "Syntax for LDIF and Command-Line Tools"

The following table lists the task areas you can manage by using Oracle Directory Manager and where to find instructions for each area.

Table 4–5 Task Areas in Oracle Directory Manager

Task Area	Instructions
Access Control Management	"Managing Access Control by Using Oracle Directory Manager" on page 14-18
	Managing Access Control by Using Command-Line Tools on page 14-48
Attribute Uniqueness Management	Chapter 8, "Attribute Uniqueness in the Directory"
Audit Log Management	Chapter 10, "Logging, Auditing, and Monitoring the Directory"

Table 4–5 (Cont.) Task Areas in Oracle Directory Manager

Task Area	Instructions
Change Log Management	"Change Logs in Directory Replication" on page 24-19 Chapter 25, "Oracle Directory Replication Administration" "Oracle Directory Synchronization Service" on page 32-6 "Synchronization Scenarios" on page 33-3 "Managing the Oracle Directory Integration and Provisioning Server" on page 35-6
Entry Management	"Managing Entries by Using Oracle Directory Manager" on page 7-2
Garbage Collection Management	Chapter 22, "Garbage Collection in Oracle Internet Directory"
Password Policy Management	Chapter 15, "Password Policies in Oracle Internet Directory"
Password Verifier Management	Chapter 16, "Directory Storage of Password Verifiers"
Plug-in Management	Part VIII, "Directory Plug-ins"
Replication Management	Chapter 25, "Oracle Directory Replication Administration"
Schema Management	"Object Classes in the Directory" on page 6-3 "Attributes in the Directory" on page 6-11
Server Management	Chapter 5, "Oracle Directory Server Administration"

Using Command-Line Tools

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

- LDAP tools, for altering objects in text files written in the LDAP Data Interchange Format (LDIF)
- A catalog management tool, for making existing attributes indexable
- Various tools to help you synchronize multiple directories in your enterprise

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

Note: To use the command-line tools, set the following environment variables:

- *ORACLE_HOME*
- ORACLE_*SID* or a proper TNS CONNECT string
- NLS_LANG (*APPROPRIATE_LANGUAGE*.AL32UTF8). The default language set at installation is AMERICAN_AMERICA.
- PATH and CLASSPATH. In the PATH and CLASSPATH environment variables, specify the Oracle LDAP binary—that is, *ORACLE_HOME*/bin—before the UNIX binary directory.

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

This section contains these topics:

Command-Line Tools for Starting, Stopping, and Monitoring Oracle Internet Directory Servers

Table 4–6 lists and describes the various command-line tools for starting, stopping, and monitoring Oracle Internet Directory servers and points you to more information about each one.

Table 4–6 Tools for Starting, Stopping, and Monitoring Oracle Internet Directory Servers

Tool	Description	More Information
OID Control Utility (OIDCTL)	Use this tool to the start and stop the server. The commands are interpreted and executed by the OID Monitor process.	"Oracle Internet Directory Architecture" on page 2-14 for a conceptual description "The OID Control Utility (oidctl) Syntax" on page A-6 for syntax and usage notes
OID Monitor (OIDMON)	Use this tool to initiate, monitor, and terminate the LDAP server processes. If you install a replication server, then 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.	"Oracle Internet Directory Architecture" on page 2-14 for a conceptual description "The OID Monitor (oidmon) Syntax" on page A-4 for syntax and usage notes

Command-Line Tools for Managing Entries and Attributes

Table 4–7 lists and describes the command-line tools for managing entries and attributes, and points you to further information.

Table 4–7 Tools for Managing Entries

Tool	Description	More Information
Catalog Management Tool (catalog.sh)	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. Useful in creating and dropping the indexes.	"The Catalog Management Tool (catalog.sh) Syntax" on page A-19 for syntax and usage notes "Indexing an Attribute by Using Command-Line Tools" on page 6-19 "Indexing an Attribute by Using Oracle Directory Manager" on page 6-16
ldapadd	Use this tool to add entries one at a time.	"ldapadd Syntax" on page A-21
ldapaddmt	Use this tool to add several entries concurrently by using this shared-server tool.	"ldapaddmt Syntax" on page A-23
ldapbind	Use this tool to authenticate user/client to a directory server.	"ldapbind Syntax" on page A-25
ldapcompare	Use this tool to see whether an entry contains a specified attribute value.	"ldapcompare Syntax" on page A-26
ldapdelete	Use this tool to delete entries.	"ldapdelete Syntax" on page A-28
ldapmoddn	Use this tool to 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-30
ldapmodify	Use this tool to create, update, and delete attribute data for an entry.	"ldapmodify Syntax" on page A-31
ldapmodifymt	Use this tool to modify several entries concurrently by using this shared-server tool.	"ldapmodifymt Syntax" on page A-37
ldapsearch	Use this tool to search for directory entries.	"ldapsearch Syntax" on page A-39

Command-Line Tools for Performing Bulk Operations

Table 4–8 lists and describes the command-line tools for performing bulk operations, and points you to further information.

Table 4–8 Command-Line Tools for Performing Bulk Operations

Tool	Description	More Information
bulkdelete	Use this tool to delete a subtree efficiently	"bulkdelete Syntax" on page A-44
bulkload	Use this tool to load and append large numbers of entries to Oracle Internet Directory through LDIF files	"bulkload Syntax" on page A-45
bulkmodify	Use this tool to modify a large number of existing entries efficiently	"bulkmodify Syntax" on page A-51
ldifwrite	Use this tool to 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-53

Command-Line Tools for Managing Replication

Table 4–9 lists and describes the command-line tools for managing replication, and points you to further information.

Table 4–9 Command-Line Tools for Managing Replication

Tool	Description	More Information
Replication Environment Management Tool	This tool ensures that Oracle9*i* Advanced Replication is properly configured for directory replication. In the event of a directory replication failure, this tool looks for the problems and seeks to rectify them. If it cannot solve the problem, then it gives you a report of the nature of the problem and points you to a possible solution.	"The Replication Environment Management Tool" on page A-62 for syntax and examples
OID Reconciliation Tool	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 on the consumer with those on the supplier by using the OID Reconciliation Tool **3.** Place the change either back into the retry queue or into the purge queue	"About the OID Reconciliation Tool" on page 25-22 "The OID Reconciliation Tool" on page A-59 for syntax and an explanation of how OID Reconciliation Tool works
Human Intervention Queue Manipulation Tool	Once you have reconciled conflicting changes by using the OID Reconciliation Tool, the Human Intervention Queue Manipulation Tool enables you to move them 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 change log entry.	"About the Human Intervention Queue Manipulation Tool" on page 25-21 "The Human Intervention Queue Manipulation Tool" on page A-56 for syntax

Command-Line Tools for Managing Directory Synchronization and Provisioning

Table 4–10 lists and describes the command-line tools for managing directory synchronization and provisioning, and points you to further information.

Table 4–10 Command-Line Tools for Managing Directory Synchronization and Provisioning

Tool	Description	More Information
Directory Integration and Provisioning Assistant	This tool assists you in performing all operations in the Oracle Directory Integration and Provisioning platform.	"The Directory Integration and Provisioning Assistant" on page A-107
Provisioning Subscription Tool	Use this tool to administer provisioning profile entries in the directory, including creating, disabling, enabling, deleting, monitoring, and clearing errors	"The Provisioning Subscription Tool (oidprovtool) Syntax" on page A-127
ldapuploadagentfile.sh	Use this tool to load mapping and configuration information when you are synchronizing directories.	"The ldapUploadAgentFile.sh Tool Syntax" on page A-120
ldapcreateconn.sh	Use this tool to create a synchronization profile	"The ldapCreateConn.sh Tool Syntax" on page A-121
oidmdelp	Use this tool to deregister a synchronization profile	"The ldapDeleteConn.sh Tool Syntax" on page A-123
stopodis	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	"The StopOdiServer.sh Tool Syntax" on page A-124
schemasync	Use this tool to synchronize schema elements—namely attributes and object classes—between an Oracle directory server and third-party LDAP directories	"The schemasync Tool Syntax" on page A-125

OID Migration Tool (ldifmigrator)

Use this tool to migrate data from application-specific repositories into Oracle Internet Directory.

See Also: "The OID Migration Tool (ldifmigrator) Syntax" on page A-135 for instructions on using this tool

OID Database Statistics Tool (oidstats.sh)

Use this tool to analyze the various database ods schema objects to estimate the statistics. 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.

See Also: "OID Database Statistics Collection Tool (oidstats.sh) Syntax" on page A-133

OID Database Password Utility (oidpasswd)

The OID Database Password Utility is used to:

- Change the password to the Oracle Internet Directory database.

  Oracle Internet Directory uses a password when connecting to an Oracle database. The default for this password matches the value you specified during installation for the Oracle Application Server administrator's password. You can change this password by using the OID Database Password Utility.

- Create a wallet, named oidlpwdldap1, for the Oracle Internet Directory database password, and a wallet, named `oidpwdr`*`sid`*, for the Oracle directory replication server password.

  The *`sid`* is obtained not from the environment variable *`SID`* but from the connected database.

  With the `create_wallet=true` option, you need to provide the ODS password to authenticate yourself to the ODS database before the ODS wallet can be generated. Note that the default ODS password is the same as that for the Oracle Application Server administrator.

- Unlock a locked directory super user account, namely, `cn=orcladmin`.

See Also: "OID Database Password Utility (oidpasswd) Syntax" on page A-131

Routine Administration at a Glance

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

Table 4–11 Routine Administration 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 6-17
Add, modify, or delete an attribute by using the Oracle Directory Manager	"Attributes in the Directory" on page 6-11
Managing Entries	-
Add, modify, or delete a directory entry by using command-line tools	"Managing Entries by Using Command-Line Tools" on page 7-10
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-45 "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 6-9
Add, modify, or delete object classes by using Oracle Directory Manager	"Object Classes in the Directory" on page 6-3
Managing Replication	-
Set up replication	Chapter 25, "Oracle Directory Replication Administration"
Resolve replication change conflicts	"Resolving Conflicts Manually in a Multimaster Replication Group" on page 25-20
Move replication changes from human intervention queue to either the retry queue or the purge queue	"About the Human Intervention Queue Manipulation Tool" on page 25-21

Table 4–11 (Cont.) Routine Administration Tasks

Task	Information
Managing Security	-
Set up an Access Control Policy Point (ACP)	Chapter 14, "Directory Access Control"
Set up SSL	Chapter 13, "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 5-7
Configure server instance parameters by using 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 by Using Oracle Directory Manager" on page 4-3 "Connecting to Additional Directory Servers by Using Oracle Directory Manager" on page 4-11
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 5-9

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
- Viewing Active Server Instance Information
- Closing Idle LDAP Connections
- Changing the Password to the Oracle Internet Directory Database Server
- Dereferencing Alias Entries
- Locating Directory Servers in a Distributed Environment

> **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.

This section contains these topics:

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

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 3-2 for instructions on how to start the server by using OID Control Utility

Preliminary Considerations for Managing Configuration Set Entries

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

To change values that should not be in effect for every server instance, it is better to create new configuration set entries. Note that this applies to the Oracle directory server and Oracle directory integration 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, then 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

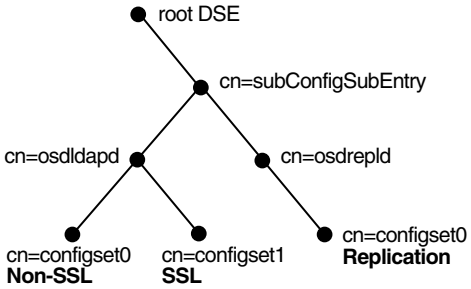


Figure 5–1 shows:

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

Note: You can run multiple instances if the directory server is on the same computer. For example, you can run one instance in SSL mode and another in non-SSL mode.

See Also:

- Chapter 13, "Secure Sockets Layer (SSL) and the Directory" for information about configuration parameters for SSL
- Chapter 25, "Oracle Directory Replication Administration" for information about configuration parameters for replication
- "Configuration Set Entry Schema Elements" on page B-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. Instead, 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 do not affect the older instances that are still running, however, unless they are restarted.

For information on restarting directory server instances, see "Restarting Oracle Internet Directory Server Instances" on page A-16.

Viewing Configuration Set Entries by Using Oracle Directory Manager

To view configuration set entries:

1. In the navigator pane, expand each of the following objects in succession: Oracle Internet Directory **Servers**, ***directory server instance***, **Server Management**.
2. Select **Directory Server**, **Replication Server**, or **Integration Server**. The parameters of the active instance appear in the right pane.
3. In the right pane, select an instance, then choose **View Properties**. 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 these parameters 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 5-6

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 for the new configuration set entry, then copy from it 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 each of the following objects in succession: Oracle Internet Directory **Servers**, ***directory server instance***, **Server Management**, **Directory Server**.
2. Select **Default Configuration Set**.
3. On the toolbar, choose **Create Like**. The Configuration Sets dialog box displays the **General** tab page.
4. In the **General** tab page, fill in the fields. These are described in Table C–32 on page C-27.
5. Select the **SSL Settings** tab and fill in the fields. These are described in Table C–33 on page C-28.
6. Choose **Apply**.
7. Restart the server instance for the command to take effect.

See Also:

- "Restarting Oracle Internet Directory Server Instances" on page A-16
- *Oracle Advanced Security Administrator's Guide* for instructions on using the Oracle Wallet Manager to set the location of the Oracle Wallet and the Oracle Wallet password
- "Setting Debug Logging Levels" on page 10-6

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 each of the following objects in succession: Oracle Internet Directory **Servers**, ***directory server instance***, **Server Management**, **Directory Server**.

2. Select **Default Configuration Set**.
3. On the toolbar, choose **Create**. A Configuration Sets dialog box displays the **General** tab page.
4. In the **General** tab page, fill in the fields. These are described in Table C–32 on page C-27.
5. Select the **SSL Settings** tab and fill in the fields. These are described in Table C–33 on page C-28.
6. Choose **OK**.

Modifying Configuration Set Entries by Using Oracle Directory Manager

To modify configuration set entries:

1. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers**, ***directory server instance***, **Server Management**, **Directory Server**.
2. Select the configuration set entry you want to modify. The configuration set appears in the group of tab pages in the right pane.
3. In the **General** tab page, modify the fields. These are described in Table C–32 on page C-27. To save the changes, choose **Apply**.
4. Select the **SSL Settings** tab and modify the fields. These are described in Table C–33 on page C-28. To save the changes, choose **Apply**.
5. Restart the server instance for the command to take effect.

See Also:

- "Restarting Oracle Internet Directory Server Instances" on page A-16
- *Oracle Advanced Security Administrator's Guide* for instructions on using the Oracle Wallet Manager to set 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 each of the following objects in succession: **Server Management**, **Directory Server**.
2. Select the configuration set entry you want to delete.

3. On the toolbar, choose **Delete**.
4. Restart the server instance for the command to take effect.

> **See Also:** "Restarting Oracle Internet Directory Server Instances" on page A-16

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 must be written in **LDAP Data Interchange Format (LDIF)**. It must 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, then 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 and the wallet location is: `/HOME/test/wallet`.

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

2. Run ldapadd with an input file.

   At the system prompt, type the command to add the input file.

   ```
   ldapadd [options] -f LDIF_file_name
   ```

   See Also:

   - "LDAP Data Interchange Format (LDIF) Syntax" on page A-2
   - "ldapadd Syntax" on page A-21 for a detailed list of options available with this command
   - "Configuration Set Entry Schema Elements" on page B-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 next example, 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.

   ```
   ldapmodify [options] -f LDIF_file_name
   ```

   See Also:

   - "LDAP Data Interchange Format (LDIF) Syntax" on page A-2
   - "ldapmodify Syntax" on page A-31 for a more detailed discussion of ldapmodify, and a list of its options
   - "Configuration Set Entry Schema Elements" on page B-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 super user 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 directory server. System operational attributes appear in the right pane.

Table C–34 on page C-29 describes the fields displayed in Oracle Directory Manager for system operational attributes.

Setting System Operational Attributes by Using ldapmodify

To modify system operational attributes, use ldapmodify. Table B–34 on page B-41 describes the modifiable system operational attributes.

> **See Also:** "ldapmodify Syntax" on page A-31 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. This section contains these topics:

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

To publish a naming context, 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 in the `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

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 choose **Browse** to open a search window.
3. Choose **Apply**.

Publishing Naming Contexts by Using ldapmodify

The following sample LDIF 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 12-5, is typically used in an environment with a middle tier such as a firewall, an application such as Oracle Delegated Administration Services, 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.

> **See Also:** Chapter 14, "Directory Access Control" for information on how to set access rights

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.

This section contains these topics:

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

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

Note: The passwords for super users, 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**, then select a directory server. The group of tab pages for that server appear in the right pane.
2. 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.
3. Edit the appropriate field in the **System Passwords** tab page as described in Table C–35 on page C-33. To save your changes, choose **Apply**.

Managing Super Users, Guest Users, and Proxy Users by Using ldapmodify

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

Table 5–1 Names, Passwords, and Attributes for Super, Guest, and Proxy Users

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

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-31 for ldapmodify syntax and usage notes.

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 choose **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

Closing Idle LDAP Connections

You can specify the number of minutes that LDAP connections remain idle before closing. To do this, you set a value for the `orclLDAPconnTimeout` attribute described in Table B–34 on page B-41.

Changing the Password to the Oracle Internet Directory Database Server

The Oracle Internet Directory uses a password when connecting to its own designated Oracle database. The default for this password when you install Oracle Internet Directory is the same as that for the Oracle Application Server administrator. You can change this password by using the **OID Database Password Utility**.

> **See Also:** "OID Database Password Utility (oidpasswd) Syntax" on page A-131

Dereferencing Alias Entries

Because entries sometimes have distinguished names that are fairly long and cumbersome, Oracle Internet Directory makes it easier to administer them by using alias objects. When someone looks up—that is, references—an object by using an alias, the alias is dereferenced, and what is returned is the object to which the alias points. For example, the alias, `Server1`, can be dereferenced so that it points to the fully qualified DN—namely, `dc=server1,dc=us,dc=myCompnay,dc=com`. This feature also enables you to devise structures that are not strictly hierarchical.

This section provides examples of how to add, search for, and modify alias entries, and it includes a list of messages. It contains these topics:

- About Alias Entries
- Examples: Using Alias Entry Dereferencing

- Success and Error Messages

About Alias Entries

An alias entry uses the object class `alias` to distinguish it from object entries in a directory. The definition of that object class is as follows:

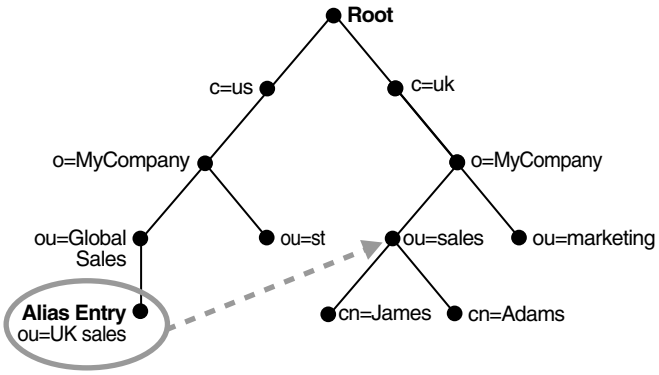
```
(2.5.6.1 NAME 'alias' SUP top STRUCTURAL MUST aliasedObjectName)
```

An alias entry also contains the `aliasedObjectName` attribute that, in turn, contains the DN of the object to which it is pointing. The definition of that attribute is as follows:

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

Figure 5–2 and the accompanying text provides an example of alias entry dereferencing.

Figure 5–2 Alias Entries Example



In Figure 5–2, `ou=uk sales,ou=global sales,o=myCompany,c=us` is an alias entry pointing to the `ou=sales,o=myCompany,c=uk` entry.

When anyone references `ou=uk sales,ou=global sales,o=oracle,c=us`, the directory server automatically reroutes them to the real entry, `ou=sales,o=oracle,c=uk`.

Examples: Using Alias Entry Dereferencing

This section contains these examples:

- Example: Adding an Alias Entry
- Examples: Searching the Directory with Alias Entries
- Example: Searching One-Level
- Example: Searching a Subtree
- Example: Modifying Alias Entries

Example: Adding an Alias Entry

To add an alias entry, you create a normal entry in LDIF and an alias entry pointing to the real entry. Following the steps in this example produces the tree in Figure 5–3 on page 5-17.

1. Create a sample LDIF file, My_file.ldif, 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
aliasedobjectname:  cn=John Doe, o=oracle, c=us
```

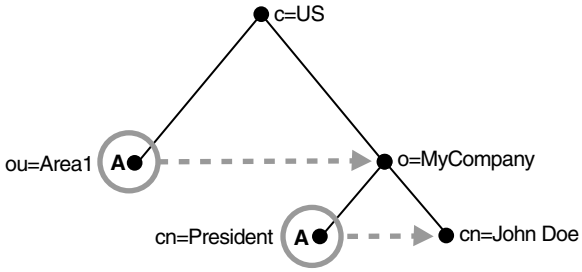
2. Add these entries to the directory by using the following command:

```
ldapadd -p port -h host -f  My_file.ldif
```

Note: When you add an alias entry whose parent is an alias entry, the directory server returns an error.

See Also: Entry Alias Dereferencing Messages on page 5-19 for error messages

Figure 5–3 Resulting Tree when Creating the My_file.ldif



In Figure 5–3, the letter A represents an alias entry, where:

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

Examples: Searching the Directory with Alias Entries

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

The flags pertaining to alias dereferencing are `-a never` and `-a find`.

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

Example: Searching the Base A base search finds the top level of the alias entry you specify.

This example shows a base search of `ou=Area1,c=us` with a filter of `"objectclass=*"` with the dereferencing flag set to `-a find`.

```
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. However, if the base is an alias entry and,

as in the example, `-a find` is specified in the search request, then the directory server automatically dereferences the alias entry and returns the entry it points to. In this example, the search dereferences `ou=Area1,c=us`, which is an alias entry, and returns `o=MyCompany,c=us`.

Example: Searching One-Level A one-level search finds only the child to the base level you specify.

This example shows a one-level search of `"ou=Area1,c=us"` with a filter of `"objectclass=*"` with the dereferencing flag set to `-a find`.

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

The directory server performs the search in two steps.

1. It searches for the base that is specified in the search request.
2. When it locates the base, 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, and thus the directory 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=MyCompany,c=us`. One of the one-level entries is `cn=President,o=MyCompany,c=us` that is not dereferenced and is returned as is.

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

Example: Searching a Subtree A subtree search finds the base, children, and grand children.

This example shows a subtree search of `"ou=Area1,c=us"` with a filter of `"objectclass=*"` with the dereferencing flag set to `-a find`.

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

The directory server performs the search in two steps.

1. It searches for the base that is specified in the search request.
2. When it 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, and thus the directory 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=MyCompany,c=us`. One of the entries is `cn=President,o=MyCompany,c=us` that is not dereferenced and is returned as is.

Thus, the search returns:

- `o=MyCompany,c=us`
- `cn=john doe,o=MyCompany,c=us`
- `cn=President,o=MyCompany,c=us`

Example: Modifying Alias Entries

This example shows how to modify alias entries. It creates a sample LDIF file, `My_file.ldif` with following entries:

```
dn: cn=President, o=MyCompany, c=us
changetype : modify
replace: aliasedobjectname
aliasedobjectname:  cn=XYZ, o=MyCompany, c=us
```

Modify the alias entry using the following command:

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

Success and Error Messages

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

Table 5–2 Entry Alias Dereferencing Messages

Message	Meaning
Alias Problem	Either of the following have occurred: ■ 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 user cannot dereference an alias because of access control issues.

Table 5–2 (Cont.) Entry Alias Dereferencing Messages

Message	Meaning
No Such Object	The server cannot 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 directory server returns this error message to the client.
Success	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 user does not have access to the dereferenced entry.

Locating Directory Servers in a Distributed Environment

To perform an operation on a particular entry, a client must be able to find the server in which that entry resides. In a distributed environment, information about the location of a server can be available in one of two ways:

- Statically, in the directory server usage file (`ldap.ora`) stored on the client host
- Dynamically, by using the domain name system (DNS). In this case, the information about server location is stored and managed in a central domain name server. The client, at request processing time, retrieves this information from the domain name server dynamically.

This section discusses these two methods of locating server information. It contains these topics:

- Static Directory Server Discovery by Using the Directory Server Usage File (ldap.ora)
- Dynamic Directory Server Discovery by Using the Domain Name System (DNS)

> **See Also:** "Discovering LDAP Services with DNS," Michael P. Armijo *et alii* (draft-ietf-ldapext-locate-08.txt) at `http://www.ietf.org`
>
> "A DNS RR for specifying the location of services (DNS SRV)", Internet RFC 2782 at `http://www.ietf.org`

Static Directory Server Discovery by Using the Directory Server Usage File (ldap.ora)

Using this method, when a client seeks to perform an operation on a directory entry, it obtains directory server location information from the directory server usage file (`ldap.ora`) stored on the client host. This file contains configuration parameters that specify:

- The type of directory server—for example, Oracle Internet Directory, Microsoft Active Directory, SunONE Directory Server, and so forth
- The location of the directory server
- The default directory entry that the client or server will use to look up or configure connect identifiers for connections to database services

The file `ldap.ora` resides in the file system of the LDAP client. When the client looks for this file, it follows this precedence:

- First, it looks at the file system directory pointed to by the `LDAP_ADMIN` environment variable
- Then it looks at the directory *`ORACLE_HOME`*`/ldap/admin` (or, on Microsoft Windows NT, *`ORACLE_HOME`*`\ldap\admin`)
- Then it looks at the file system directory pointed to by the `TNS_ADMIN` environment variable
- Finally, it looks at the directory *`ORACLE_HOME`*`/network/admin` (or, on Microsoft Windows NT, *`ORACLE_HOME`*`\network\admin`)

If the file `ldap.ora` is present in more than one location, then the location having higher precedence is honored.

Using the static method to discover a directory server can increase management overhead. For example, because the `ldap.ora` file is stored on the client host, the administrator must update that file on every client whenever the host name or port number of a directory server is changed. To avoid this increased overhead, you can enable an application to discover directory servers dynamically by using the domain name system (DNS).

Dynamic Directory Server Discovery by Using the Domain Name System (DNS)

The domain name system (DNS) is a dynamic way of locating domain names and translating them into the actual addresses of computers. This translation process is handled by a central domain name server, which contains information about the locations of directory servers.

Once a network administrator has entered the necessary information about directory server locations in a domain name server, clients can retrieve that information from that server instead of from `ldap.ora` files.

For a client to locate a directory server by using DNS, the following steps must have been completed:

- The network administrator must have entered a DNS Service Location Record (SRV) into the domain name server.
- The client application must have been enabled to map distinguished names to domain names.

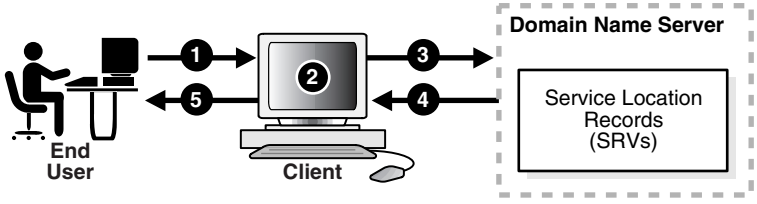
How a Client Locates a Directory Server by Using DNS

To find the directory server on which an entry resides, a client communicates with the domain name server. Specifically, it provides to the domain name server a domain name. The domain name specifies where the needed directory server is located.

To generate the domain name, the client extracts the domain component from the DN entered by the user. For example, in the DN `cn=John Doe,ou=accounting,dc=example,dc=net`, the domain component is `dc=example,dc=net`. That domain component represents the server on which the requested entry resides. The client then converts that domain name component to a domain name in a format recognized by the domain name server, namely, `example.net`.

Figure 5–4 and the accompanying text show the process of locating a directory server from the perspective of a client.

Figure 5–4 A Client Locating a Directory Server by Using DNS



1. A user wanting to perform an operation on a directory entry, enters into the client the distinguished name (DN) of that entry—for example, `cn=John Doe,ou=accounting,dc=example,dc=net`.

2. To communicate with the domain name server, the client converts the domain component of the DN to a domain name. In the example used here, the client would convert the domain component of that DN—namely, `dc=example,dc=net`—to the domain name `example.net`.
3. The client queries the domain name server for SRV resource records having the specified domain name.
4. The domain name server returns the SRV resource records that match the specified domain name. These resource records contain the host name information of the directory server containing the requested entry. If the domain name server is not able to find any matching SRV resource records, then it returns an error message.
5. The client parses the records. It extracts the directory host name information from these records and returns it to the user.

See Also:

- P. Mockapetris, Domain Names—Concepts and Facilities (RFC 1034) at `http://www.ietf.org`
- P. Mockapetris, Domain Names—Implementation and Specification (RFC 1035) at `http://www.ietf.org`

Note: The domain name server either stores all the necessary SRV records locally, or obtains them from other domain name servers. If the domain name server cannot find the requested information, then it returns an error message. It does not return a referral to another domain name server.

Registering a Directory Server with the Domain Name System

Registering server location information for a directory server involves entering a DNS service location record (SRV) into the domain name server. The SRV record contains:

- The DNS name of the server that provides the LDAP service
- The corresponding port number
- Parameters that enable the client to choose an appropriate server from multiple servers

The SRV resource record enables administrators to use several servers for a single domain, to move services from host to host easily, and to designate some hosts as primary servers for a service and others as backups.

The format of the SRV record can be either specific to Oracle Internet Directory servers or standard. For information about Oracle Internet Directory servers, the Oracle Internet Directory-specific format is preferred. When a client first queries a domain name server, it looks for SRV records that have the Oracle Internet Directory-specific format. If it does not find any with this format, then it queries for SRV records that have the standard format.

The Oracle Internet Directory-Specific Format for SRV Records

The Oracle Internet Directory-specific format is:

```
_Service._Proto._product.Domain TTL Class Type Priority Weight Port Target
```

Table 5–3 describes the arguments. The following is an example of an SRV record that uses the Oracle Internet Directory-specific format.

```
_ldap._tcp._oid.acme.com 0 IN SRV 0 1 389 ldap.acme.com
```

The Standard Format for SRV Records

The standard format is:

```
_Service._Proto.Domain TTL Class Type Priority Weight Port Target
```

Table 5–3 describes the arguments. The following is an example of an SRV record for a non-SSL-based directory server that uses the standard format.

```
_ldap._tcp.acme.com 0 IN SRV 0 1 389 ldap.acme.com
```

Table 5–3 Arguments in a Service Location Record (SRV)

Argument	Description
Service	For a non-SSL-based server, the value for this argument is `ldap`. For an SSL-based server, the value is `ldaps`.
Proto	The value is always `tcp`.
Product	The value is always `oid`.

Table 5–3 (Cont.) Arguments in a Service Location Record (SRV)

Argument	Description
Domain	The domain name. It is usually obtained by converting the DN of the naming context mastered by the directory server into a domain name. **See Also:** "How a Client Locates a Directory Server by Using DNS" on page 5-22
TTL	Time to live. This argument has the standard DNS meaning. It specifies how long the resource record may be cached before the source of the information is again consulted.
Class	This argument has the standard DNS meaning. SRV records occur in the IN class.
Type	For all SRV records, the value for this argument is SRV.
Priority	The priority of the directory server. A client must attempt to contact the target host with the lowest-numbered priority.
Weight	A server selection mechanism, this argument specifies a relative weight for entries with the same priority. If multiple SRVs have the same priority, then they are ordered according to the following protocol: **1.** To select a target to be contacted next, arrange in any order all SRV resource records that have not yet been ordered—but place all those with weight 0 at the beginning of the list. **2.** Compute the sum of the weights of those resource records, and with each resource record associate the running sum in the selected order. **3.** Choose a uniform random number between 0 and the sum computed (inclusive), and select the resource record whose running sum value is the first in the selected order that is greater than or equal to the random number selected. The target host specified in the selected SRV resource record is the next one to be contacted by the client. **4.** Remove this SRV resource record from the set of the unordered SRV resource records. **5.** Apply the described algorithm to the unordered SRV resource records to select the next target host. **6.** Continue the ordering process until there are no unordered SRV resource records. **7.** Repeat this process for each priority.

Table 5–3 (Cont.) Arguments in a Service Location Record (SRV)

Argument	Description
Port	The port on target host for the directory service.
Target	The domain name of the host on which the directory server is running.

Note: If the directory server is moved to a different host or is run on different port, then the corresponding SRV resource record must be updated accordingly.

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:

- 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—that is, metadata such as that for an **object class**, an **attribute**, a **matching rule**, and syntax. This information is stored in a special class of entry called a **subentry**. More specifically, Oracle Internet Directory, following LDAP Version 3 standards, stores this information 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.

Object Classes in the Directory

This section contains these topics:

- About Object Class Management
- Managing Object Classes by Using Oracle Directory Manager
- Managing Object Classes by Using Command-Line Tools

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 B, "Oracle Internet Directory Schema Elements" for a list of schema elements installed with Oracle Internet Directory

Guidelines for Adding Object Classes

When you add a directory entry, you associate it with one or more object classes. Each object class contains attributes that you want to associate with the new entry. For example, if you are creating an entry for an employee, you can associate the entry with the `person` object class. This object class contains many of the attributes that you want to associate with that employee entry—including, for example, name, address, and telephone number.

Inheritance Each object class derives from a hierarchy of superclasses, and it inherits attributes from these superclasses. By default, all object classes inherit from the `top` object class. When you assign an object class to an entry, the entry inherits all of the attributes of both that object class as well as its superclasses.

Mandatory and Optional Attributes in Object Classes The attributes that entries **inherit** from an super class may be either mandatory or optional. Values for 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 6-11

Addition of Entries in Top-Down Sequence Entries must be added in a top-down sequence—that is, 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 because the directory server is delivered with a full set of standard directory objects.

Object Class Explosion 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, 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.

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.

See Also: "Subclasses, Superclasses, and Inheritance" on page 2-8 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.
- If existing object classes do not have the attributes you need, you can create an auxiliary object class and associate the needed attributes with that object class.
- 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:

- "Object Classes in the Directory" on page 6-3
- "Managing Object Classes by Using Command-Line Tools" on page 6-9

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 tells you how to use Oracle Directory Manager to search for object classes, view their properties, add, modify, and delete them.

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, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select Schema Management. The Schema Management tab pages appear in the right pane.

3. Choose the **Find Object Classes** button at the lower right of the right pane, or, from the menu bar, choose **Edit**, then choose **Find Object Classes**. The Find: Object Classes dialog box appears.
4. 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 listed and described in Table C–20 on page C-18.

> **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.

5. In the menu in the middle of the search criteria bar, select the filter you want to use for your search. Options are listed and described in Table C–21 on page C-18.
6. In the text box at the right end of the search criteria bar, type the value of the property of the object class you are searching for. For example, to search for all object classes with names that begin with the letters `orcl`, type those letters in the text box at the right end of the search criteria bar.
7. Below the **Criteria** field are five buttons described in the next table. Use these buttons to further refine your search.
8. Choose **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 each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**.
3. 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, choose the object class. The properties of the selected object class appear in the Object Class dialog box.

Adding Object Classes by Using Oracle Directory Manager

To add object classes by using Oracle Directory Manager:

1. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**.
3. In the right pane, select the **Object Classes** tab and choose the **Create** button in the toolbar. The New Object Class dialog box appears.

   Alternatively, in the **Object Classes** tab page, select an object class that is similar to one you would like to create, and then choose **Create Like**. The New Object Class dialog box displays the attributes of the selected object class. You can create the new object class by using the selected one as a template.
4. In the New Object Class dialog box, enter the information in the fields. These are described in Table C–23 on page C-20.
5. Choose **OK**.

See Also:

- "Object Class Types" on page 2-9
- "Subclasses, Superclasses, and Inheritance" on page 2-8
- 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, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**.
3. In the right pane, select the **Object Classes** tab and choose the object class you want to modify. The Object Class dialog box appears.
4. In the Object Class dialog box, modify or add the information in the fields. These are described in Table C–23 on page C-20.
5. Choose **OK**.

See Also:

- "Object Class Types" on page 2-9
- "Subclasses, Superclasses, and Inheritance" on page 2-8

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 6-10 for an example of adding attributes to an auxiliary object class

---

Deleting Object Classes by Using Oracle Directory Manager

Caution: Oracle Corporation recommends that you not delete object classes from the base schema. If you delete an object class that is referenced by any entries, those entries then become inaccessible.

Should you decide to delete an object class from the base schema, be careful not to delete one that is in use or that you might want to use in the future.

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 page and select the object class you want to delete.
3. Choose **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-31

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 previous 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 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 )  )
```

Attributes in the Directory

This section contains these topics:

- About Attribute Management
- Managing Attributes by Using Oracle Directory Manager
- Managing Attributes by Using Command-Line Tools

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 7-8 and "Managing Entries with Attribute Options by Using Command-Line Tools" on page 7-12 for instructions on adding and deleting attribute options and for searching for entries containing attribute options
- "Size of Attribute Values" on page B-47 for information about using syntax to specify the size of the attribute value

About Attribute Management

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 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.

> **See Also:** "Size of Attribute Values" on page B-47 for information about using syntax to specify the size of the attribute value

Managing Attributes by Using Oracle Directory Manager

This section tells you how to use Oracle Directory Manager to search for, view, add, modify, delete, and index attributes.

Viewing All Directory Attributes by Using Oracle Directory Manager

To view attributes by using Oracle Directory Manager:

1. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers**, ***directory server instance***.
2. Select **Schema Management**.
3. In the right pane, select the **Attributes** tab page This tab page displays a table containing the attribute properties. The columns in this table are described in Table C–24 on page C-20.

> **See Also:** "Viewing Attributes for a Specific Entry by Using Oracle Directory Manager" on page 7-4 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, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**. The corresponding tab pages appear in the right pane.
3. Select the **Attributes** tab page.
4. Choose the Find **Attributes** button in the lower right corner. The Find Attributes dialog box appears
5. 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 described in Table C–24 on page C-20.
6. In the menu in the middle of the search criteria bar, select the filter you want to use for your search. Options are described in Table C–25 on page C-21.
7. 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`.

8. To further refine your search, use the buttons in the **Search Criteria** box to enhance the search criteria bar. These are described in Table C–26 on page C-22.
9. Choose **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. See "Creating a New Attribute from an Existing One by Using Oracle Directory Manager" on page 6-14.

Adding a New Attribute by Using Oracle Directory Manager To add a new attribute:

1. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**.
3. In the right pane, select the **Attributes** tab, then choose the **Create** button in the toolbar. 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.
4. In the **General** tab, enter values in each of the fields. These are described in Table C–27 on page C-22.
5. Select the **Advanced** tab, and enter values in each of the fields. These are described in Table C–28 on page C-23.
6. Choose **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 6-5.

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, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**.
3. In the right pane, select the **Attributes** tab.
4. In the **Attributes** tab page, select the attribute you want to copy.
5. Choose **Create Like**. The New Attribute Type dialog box for that attribute appears. This dialog box contains two tab pages—**General** and **Advanced**.
6. Select the **General** tab and enter values in each of the fields. These are described in Table C–27 on page C-22. You must always change the DN to that of the new attribute.
7. Select the **Advanced** tab and enter values in each of the fields. These are described in Table C–28 on page C-23.
8. Choose **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 6-5.

Modifying an Attribute by Using Oracle Directory Manager

To modify an attribute by using Oracle Directory Manager:

1. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**.
3. In the right pane, select the **Attributes** tab, then select an editable attribute in the list.
4. Choose **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.
5. Select the **General** tab and enter values in each of the fields. These are described in Table C–27 on page C-22.

6. Select the **Advanced** tab and enter values in each of the fields. These are described in Table C–28 on page C-23.
7. Choose **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, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**.
3. In the right pane, select the **Attributes** tab, then select an editable attribute in the list.
4. Choose **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 6-19.

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 B-47
- Less than 28 characters in their names

Viewing Indexed Attributes by Using Oracle Directory Manager To view indexed attributes:

1. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**.
3. In the right pane, select the **Attributes** tab page. This tab page 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 To add an index to an attribute:

1. Create an attribute as described in "Adding an Attribute by Using Oracle Directory Manager" on page 6-14.
2. In the New Attribute Type dialog box, on the **Advanced** tab page, 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, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**.
3. In the right pane, select the **Attributes** tab.
4. 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.
5. Choose **Drop Index**.

Managing Attributes by Using Command-Line Tools

This section discusses adding, modifying, and indexing attributes 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_file_name
```

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' )
```

To specify an attribute as single-valued, include in the attribute definition entry in the LDIF file the keyword SINGLE-VALUE with surrounding white space.

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-31 for a detailed explanation of ldapmodify and its options
- "Syntaxes in the Directory" on page 6-27 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_file_name
```

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-31 for a detailed explanation of ldapmodify and its options
- "Syntaxes in the Directory" on page 6-27 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 B-47
- 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-31

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:** "The Catalog Management Tool (catalog.sh) Syntax" on page A-19

Note: Unless you are absolutely sure that the indexes were not created by the base schema that was installed with Oracle Internet Directory, be careful not to use the catalog.sh `-delete` option to remove indexes from attributes. Removing indexes from base schema attributes can adversely impact the operation of Oracle Internet Directory.

How to Extend the Number of Attributes Associated with Entries

You can extend the number of attributes for entries. The method you use depends on whether the entries already exist.

For an existing entry, there are two ways to extend the attributes associated with it. One way is to add names of object classes to the list in the `objectclass` attribute for each entry. If your directory is relatively small, then this can be a desirable method because it enables searches for entries based on that attribute. However, if your directory is large, then entering the names of object classes to the `objectclass` attribute can be very painstaking. In this case, the second way, namely, using content rules, may be a more efficient way to extend the content of entries.

This section contains these topics:

Extending the Number of Attributes Prior to Creating Entries in the Directory

At installation, Oracle Internet Directory provides standard LDAP object classes and 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:

- Define a new (base) object class
- Define an object subclass

See Also:

- Appendix B, "Oracle Internet Directory Schema Elements" for a list of object classes in the schema installed with Oracle Internet Directory
- About Object Class Management on page 6-3 for instructions on how to define a new object class or object subclass

Extending the Number of Attributes for Existing Entries by Creating an Auxiliary Object Class

You can create an auxiliary object class containing the additional attributes you want for your entry, and then associate that auxiliary object class with the entry. You associate the auxiliary object class with the entry by specifying it in the `objectclass` attribute for the entry.

See Also:

- "About Object Class Management" on page 6-3 for instructions on creating auxiliary object classes
- Chapter 7, "Directory Entries Administration" for instructions on associating an object class with an entry

Extending the Number of Attributes for Existing Entries by Creating a Content Rule

A content rule, following your specifications, determines the kind of content allowed in any entry that is associated with a particular structural object class. For example, you can specify that any entry associated with the `person` object class must have, in addition to the attributes in that object class, other attributes as well. The additional attributes can be those of an auxiliary object class, and they can be either mandatory or optional. You can also specify that such entries must not contain values for one or more particular attributes.

Whereas you must list auxiliary classes in the entry—which can be an administrative burden—you do not need to list content rules in the entry.

In addition to the structural object class to which it applies, a content rule can also indicate:

- Auxiliary object classes allowed for entries governed by the rule
- Mandatory attributes, in addition to those called for by the structural and auxiliary object classes, required for entries governed by the DIT content rule
- Optional attributes, in addition to those called for by the structural and auxiliary object classes, permitted for entries governed by the DIT content rule;
- Optional attributes from the entry's structural and auxiliary object classes that are precluded from appearing in entries governed by the rule

Rules for Creating and Modifying Content Rules

Content rules are defined as values of the `DITContentRule` attribute in the subschema subentry (`cn=subschemasubentry`). They must conform to these rules:

- The structural object class of the entry identifies the content rule applicable for the entry. If no content rule is present for a structural object class, then entries associated with that object class contain only the attributes permitted by the structural object class definition.
- Because a content rule is associated with a structural object class, all entries of the same structural object class have the same content rule regardless of the their location in the DIT
- The content of an entry must be consistent with the object classes listed in the `objectClass` attribute of that entry. More specifically:
  - Mandatory attributes of object classes listed in the `objectClass` attribute must always be present in the entry

- Optional attributes of auxiliary object classes indicated by the content rule can also be present even if the `objectClass` attribute does not list these auxiliary object classes.

  See Also: "Managing Content Rules" on page 6-24 for instructions on creating and managing content rules

Schema Enforcement When Using Content Rules

When validating an object for schema consistency, the directory server uses the content rule for the structural object class of the entry. It also uses all the other object classes listed in the entry.

If more than one content rule exists for an object class, then, when adding or modifying an entry, or when bulkloading data, the following rules apply.

- An entry can have attributes from all the auxiliary object classes listed in the various content rules. Not specifying an object class in the content rule does not restrict a client from explicitly adding an auxiliary object class in directory entries.
- An entry must contain values for all the mandatory attributes listed in:
  - The content rules
  - The object classes associated with the entry
  - The auxiliary object classes listed in the content rule applicable to the entry
- Optionally, an entry can contain values for any or all the optional attributes listed in:
  - The content rule
  - The object classes listed in the entry
  - The auxiliary object classes listed in the content rule applicable for the entry
- If any attribute is specified as mandatory, then it overrides any other definition that defines it as optional.

Searches for Object Classes Listed in Content Rules

Because the auxiliary object classes listed in content rules are not listed in the `objectclass` attribute for an entry, you cannot list those object classes as filters when you search for entries. Instead, base your searches on the structural object class that you are interested in. If you need to base your search on an auxiliary

object class, then add that auxiliary object class to the `objectclass` attribute in the user objects explicitly.

For example, a content rule for structural object class `inetOrgPerson` may specify an auxiliary object class `orclUser`. However, this does not mean that every `inetOrgPerson` entry in the directory contains `orclUser` as a value of the `objectclass` attribute. As a result, the search with the filter `objectclass=orclUser` fails. Instead of querying for an auxiliary object class contained in the content rule, you should query for structural object classes—for example, `objectclass=inetOrgPerson`.

To base a search on `objectclass=orcluser`, add `orclUser` as one of the values of `objectclass` attribute in each entry.

These considerations apply also to filters used in access control policies. If you are using a content rule to associate additional auxiliary object classes, then use only the structural object classes in the search filters.

Managing Content Rules

This section tells you how to manage content rules by using Oracle Directory Manager and command-line tools.

Managing Content Rules by Using Oracle Directory Manager This section tells you how to use Oracle Directory Manager to create and modify content rules.

Creating a Content Rule by Using Oracle Directory Manager

To create a content rule:

1. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**.
3. In the right pane, select the **Content Rules** tab.
4. Choose **Create**. The New Content Rule dialog box appears.
5. In the New Content Rule dialog box, enter values in the appropriate fields. These fields are described in Table C–30 on page C-24.
6. Choose **OK**.

Modifying a Content Rule by Using Oracle Directory Manager

To modify a content rule:

1. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**.
3. In the right pane, select the **Content Rules** tab.
4. Select the content rule you want to modify, then choose **Edit**. The Content Rule dialog box appears.
5. In the Content Rule dialog box, enter values in the appropriate fields. The fields for this dialog box are described in Table C–31 on page C-25
6. Choose **OK**.

Managing Content Rules by Using Command-Line Tools The format of a content rule is:

```
DITContentRule  ::=  SEQUENCE  {
    oids                                ALPHA-NUMERIC-OID,
    structuralObjectClass      OBJECT-CLASS,
    LABEL                               CONTENT-LABELOPTIONAL,
    auxiliaries                         SET (1..MAX) OF OBJECT-CLASSOPTIONAL,
    mandatory                           SET (1..MAX) OF ATTRIBUTEOPTIONAL,
    optional                            SET (1..MAX) OF ATTRIBUTEOPTIONAL,
```

Table 6–1 describes the parameters. Note that the attribute and object class names are case-insensitive.

Table 6–1 Content Rule Parameters

Parameter	Description
oids	A unique object identifier (oids) for the content rule similar to the one for an object class or attribute defintion.It can be either numeric or alphanumeric value as long as it is unique.
LABEL	The content label of the content rule as applied in the directory
structuralObjectClass	The structural object class to which the content rule applies
auxiliaries	The auxiliary object classes allowed for an entry to which the content rule applies
mandatory	User attribute types contained in an entry to which the content rule applies. These are in addition to those mandatory attributes that the entry contains as a result of its association with its specified structural and auxiliary object classes.

Table 6–1 (Cont.) Content Rule Parameters

Parameter	Description
optional	User attribute types that may be contained in an entry to which the content rule applies. These are in addition to those that the entry may contain as a result of its association with its specified structural and auxiliary object classes.

During the process of defining a new content rule, the directory server validates the syntax and ensures that the attributes and object classes listed in the content rule have been defined in the directory.

Content rules can be specified for structural object classes only. The name of the object class is case-insensitive.

You can specify more than one content rule for each structural object class provided the content rules have different labels associated with them.

To modify an existing definition of a content rule, the client must first delete the existing definition and then add the new definition. Simple replacement of a content rule by using the `replace` command is not allowed.

To delete a content rule, the client needs to specify only the structural object class and the alphanumeric object identifier of the content rule. Optionally, the client can also specify the associated version of the content rule to be deleted.

Matching Rules in the Directory

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 each of the following objects in succession: **Oracle Internet Directory Servers** and ***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 described in Table C–29 on page C-24.

Viewing Matching Rules by Using ldapsearch

Use ldapsearch on the subentry `cn=subSchemaSubentry`.

See Also: "ldapsearch Syntax" on page A-39

Syntaxes in the Directory

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, expand each of the following objects in succession: **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Schema Management**.
3. 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-39

7

Directory Entries Administration

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, "Directory 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 each of the following objects in succession: **Oracle Internet Directory Servers**, ***directory server instance***, and **Entry Management**.

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 each of the following objects in succession: **Oracle Internet Directory Servers**, ***directory server instance***, and **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 described in Table C–37 on page C-35.

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. These are described Table C–38 on page C-36.

8. Click **Search**. The results of your search appear in the **Distinguished Name** box.

> **See Also:** "Viewing Active Server Instance Information" on page 5-13 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 6-13 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. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers**, ***directory server instance***.
2. Select **Entry Management**.
3. On the toolbar, click **Create**. The New Entry dialog box appears.
4. In the **Distinguished Name** field, type the full DN. You can 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.
5. To specify an **object class** for the new entry, next to the **Object Classes** box, click **Add**. The Super Class Selector dialog box appears.
6. 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.
7. 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 each of the following objects in succession: **Oracle Internet Directory Servers**, ***directory server instance***.
2. Select **Entry Management**.

1. 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. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers**, ***directory server instance***.
3. Select **Entry Management**.
4. On the toolbar, click **Create**. The New Entry dialog box appears.
5. In the **Distinguished Name** field, type the full DN. You can 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.
6. To the right of the Object Classes box, click **Add**. The Super Class Selector dialog box appears.
7. In the Super Class Selector dialog box, select the `person` object class, then click **Select**. This returns you to the New Entry dialog box.
8. In the New Entry dialog box, click the **Optional Properties** tab, and scroll to the User Password window.

9. Type the password for Anne Smith.

See Also:

- "Searching for Entries by Using Oracle Directory Manager" on page 7-2 for instructions on using the search pane
- "Managing Group Entries" on page 9-7
- "Security Groups" on page 14-3 for instructions on setting access control policies for group entries
- Globalization Support on page 2-13 and Chapter 14, "Directory Access Control" for information about access privileges

Modifying Entries by Using Oracle Directory Manager

You can add auxiliary object classes to an existing entry.

You can add optional, but not mandatory, attributes to an object class already in use by entries. If you add optional attributes to an object class already in use by an entry, 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. Modify the appropriate fields, then choose **Select the Properties** tab page. If you do not see the attributes you want to add or modify, then, at the top of the tab page, select **View Properties: All**.
5. In the **Properties** tab page, modify the values of any editable attributes.
6. Choose **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 7-6.

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 User Password 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 10*g* (9.0.4), 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. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers**, ***directory server instance***, and **Entry Management**.
2. Select the entry to which you want to add an attribute option. The corresponding tab pages appear in the right pane.

3. In the right pane, in the **Properties** tab page, in the **View Properties** field, select **Advanced**. The **Properties** tab page changes accordingly.
4. In the **Attribute** field, select the attribute to which you want to add the option, for example, `ou`.
5. In the **Attribute Options** field, enter the attribute option, for example, `lang-en`.
6. 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.
7. Click **Apply**.

Modifying an Attribute Option by Using Oracle Directory Manager

To modify an attribute option:

1. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers**, ***directory server instance***, and **Entry Management**.
2. Select the entry whose attribute option you want to modify. The corresponding tab pages appear in the right pane.
3. In the **Properties** tab page, in the **View Properties** field, select either **Only Non-null Values** or **All**.
4. Scroll to the field containing the attribute option you want to modify.
5. Modify the value in the field.
6. Click **Apply**.

Deleting an Attribute Option by Using Oracle Directory Manager

To delete an attribute option:

1. In the navigator pane, expand each of the following objects in succession: **Oracle Internet Directory Servers**, ***directory server instance***, and **Entry Management**.
2. Select the entry from which you want to delete an attribute option. The corresponding tab pages appear in the right pane.

3. In the **Properties** tab page, in the **View Properties** field, select either **Only Non-null Values** or **All**.
4. Scroll to the field containing the attribute option you want to delete.
5. Delete the value in the field.
6. Click **Apply**.

Managing Entries by Using Command-Line Tools

This section points you to the command-line tools you can use in managing entries. It also provides several examples of entry management by using command-line tools. It contains these topics:

- Command-Line Tools for Managing Entries
- Example: Adding a User Entry by Using ldapadd
- Example: Adding an Attribute Option by Using ldapmodify
- Example: Modifying a User Entry by Using ldapmodify
- Managing Entries with Attribute Options by Using Command-Line Tools

Command-Line Tools for Managing Entries

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

Table 7–1 Command-Line Tools for Managing Entries

Tool	Task(s)	Syntax and Usage Notes
ldapadd	Add entries one at a time. Add new configuration set entries. Configure a server with an input file.	"ldapadd Syntax" on page A-21
ldapaddmt	Add several entries concurrently by using this shared server tool.	"ldapaddmt Syntax" on page A-23
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-25
ldapcompare	Compare attribute values you specify with those in a directory entry.	"ldapcompare Syntax" on page A-26
ldapdelete	Delete entries.	"ldapdelete Syntax" on page A-28

Table 7–1 (Cont.) Command-Line Tools for Managing Entries

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-30
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-31
ldapmodifymt	Modify several entries concurrently by using this shared server tool.	"ldapmodifymt Syntax" on page A-37
ldapsearch	Search for directory entries.	"ldapsearch Syntax" on page A-39

Example: Adding a User Entry by Using ldapadd

The following example shows an LDIF file, named `entry.ldif`, for the 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 previous example, 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

Suppose that you were adding the Spanish equivalent of an entry for John, and that the data for this user entry is in the `entry.ldif` file. This file contains the following:

```
dn: cn=john,c=us
changeType: modify
add: cn;lang-sp
cn;lang-sp: Juan
```

Issue this command to modify the file:

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

Example: Deleting an Attribute Option by Using ldapmodify

The following example deletes the `cn;lang-fr` attribute option from the entry for John. As in the previous example, assume that the data for this user entry is in the `entry.ldif` file. This file contains the following:

```
dn: cn=john, c=us
changetype: modify
delete: cn;lang-fr
cn;lang-fr: Jean
```

Issue this command to modify the file:

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

Example: Searching for Entries with Attribute Options by Using ldapsearch

The following example retrieves entries with common name (`cn`) attributes that have an option specifying a language code attribute option. This particular example retrieves entries in which the common names are in French and begin with the letter R.

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub "cn;lang-fr=R*"
```

Suppose that, in the entry for John, no value is set for the `cn;lang-it` language code attribute option. In this case, the following example fails:

```
ldapsearch -p 389 -h myhost -b "c=us" -s sub "cn;lang-it=Giovanni
```

See Also: "Attribute Options" on page 2-7

Managing Entries by Using Bulk Tools

This section lists and describes some of the more common tasks you perform with bulk tools.

This section contains these topics:

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 (oidstats.sh) Syntax" on page A-133 for a description and syntax for the oidstats.sh tool
- "Using Command-Line Tools" on page 4-14 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 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

This section contains these topics:

- Task 1: Back Up the Oracle Database Server
- Task 2: Find Out the Oracle Internet Directory Password
- Task 3: Check Input for Schema and Data Consistency Violations

Task 1: Back Up the Oracle Database Server

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

> **See Also:** *Oracle9i Backup and Recovery Basics* in the Oracle9*i* Database Server 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:** "OID Database Password Utility (oidpasswd) Syntax" on page A-131

Task 3: Check Input for Schema and Data Consistency Violations

On UNIX, 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 connect_string -check path_to_ldif-file_name
```

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 connect_string -generate ldif-file_name
```

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 connect_string -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-53

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-51

Deleting a Large Number of Entries

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

See Also: "bulkdelete Syntax" on page A-44

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:** "Directory Partitioning" on page 2-26 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
- Client-Side Referral Caching

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 every **naming context** 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:** "Knowledge References and Referrals" on page 2-27 for a conceptual discussion of knowledge references

Client-Side Referral Caching

Referral caching is the process of storing referral information so that it can be easily accessed again and again. Suppose that a client queries Server A, which returns a referral to Server B. The client chases this referral and contacts Server B which performs the operation and returns the results to the client. Without referral caching, the next time the client makes the same query to Server A, the entire procedure is repeated, an unnecessary consumption of time and system resources.

However, if the referral information can be cached, then, in each subsequent query, the referral information can be obtained from cache and Server B can be contacted directly. This speeds up the operation.

Client-side referral caching enables each client to cache this referral information and use it to speed up of referral processing.

How Client-Side Referral Caching Works

Referral entries are stored in a configuration file on the client. When a client establishes a session, it reads the referral information from this configuration file and stores them in a cache. This cache remains static, with no further updates being added during the session. From this point on, for every operation, the client looks up referral information in the cache.

The directory administrator prepares this configuration file for clients to use.

> **Note:** The configuration file is optional for clients. If a file is not present, then client operations involving referrals still behave correctly. Thus it is not mandatory for administrator to prepare this file. The advantage of using the configuration file is that it speeds up the client/server operations involving referrals.

The configuration file consists of one or more referral sets. Each referral set consists of:

- The host name where a particular directory server is running
- One or more referral entries residing on that server

Each referral entry consists of a sequence of lines, each of which corresponds to one referral URL. The line separator is CR LF or LF.

```
ref_file=ref_file_content
ref_file_content=1*(referral_set)
referral_set=hostname      SEP      ref_entry_set   SEP
ref_entry_set=ref_entry     *(SEP      ref_entry)
ref_entry=1*(referralurl     SEP)
SEP=CR LF / LF
CR=0x0D
LF=0x0A
```

For example, consider two referral entries in a directory server running on host serverX:

```
dn: dc=acme, dc=com
ref: ldap://serverA:389/dc=acme, dc=com
ref: ldap://serverB:389/dc=acme, dc=com

dn: dc=oracle, dc=com
ref: ldap://serverC:389/dc=oracle, dc=com
ref: ldap://serverD:389/dc=oracle, dc=com
```

Consider the following referral entry in a directory server running on host serverY:-

```
dn: dc=fiction, dc=com
ref: ldap://serverE:389/dc=fiction, dc=com
```

The corresponding `referral.ora` file looks like this:

```
ServerX
ldap://serverA:389/dc=acme, dc=com
ldap://serverB:389/dc=acme, dc=com

ldap://serverC:389/dc=oracle, dc=com
ldap://serverD:389/dc=oracle, dc=com

ServerY
ldap://serverE:389/dc=fiction, dc=com
```

8

Attribute Uniqueness in the Directory

This chapter explains attribute uniqueness in Oracle Internet Directory. It contains these topics:

About Attribute Uniqueness

The attribute uniqueness feature prevents duplication of attribute values, both when adding and modifying them. For example, it prevents you from assigning to a new employee an identifier already assigned to another employee. Instead, the directory server terminates the operation and returns an error message.

You can define attribute uniqueness:

- Across the entire directory

  For example, to ensure that every entry in your directory that includes a `mail` attribute has a unique value for that attribute, you create an instance of attribute uniqueness associated with `mail`.

- Across one subtree for each attribute

  For example, suppose that MyCompany hosts the directories for SubscriberCompany1 and SubscriberCompany2. You can enforce attribute uniqueness in SubscriberCompany1 only.

- Across one object class

  For example, `ID` is an attribute in both the `machine` object class, and the `person` object class. If attribute uniqueness is enabled, then the directory server prevents you from adding either two machines or two people with the same `ID`. However, a `machine ID` attribute can have the same value as a `person ID` attribute.

To implement attribute uniqueness, you create an attribute uniqueness constraint entry in which the attributes in Table 8–1 on page 8-2 are specified.

Table 8–1 Attribute Uniqueness Constraint Entry

Attribute Name	Mandatory?	Valid Value	Default Value	Default Effect
`orcluniqueattrname`	Yes	Any string	N/A	N/A
`orcluniquescope`	No	One of the following: ▪ `base`—Searches the root entry only ▪ `onelevel`—Searches one level only ▪ `sub`—Searches the entire directory	`sub`	Searches the entire directory

Table 8–1 (Cont.) Attribute Uniqueness Constraint Entry

Attribute Name	Mandatory?	Valid Value	Default Value	Default Effect
orcluniqueenable	No	Either 0 (disable) or 1 (enable)	0	Disables attribute uniqueness
orcluniquesubtree	No	Any string	" "	Searches the entire directory
orcluniqueobjectclass	No	Any string	" "	Searches all object classes

When you have created the entry and specified the attributes, before it performs an operation, the directory server:

- Uses the attribute uniqueness constraint to check all update operations
- Determines whether the operation applies to a monitored attribute, subtree, or object class

If an operation applies to a monitored attribute, suffix, or object class, and would cause two entries to have the same attribute value, then the directory server terminates the operation and returns a constraint violation error message to the client.

Note: The attribute uniqueness feature works on indexed attributes only.

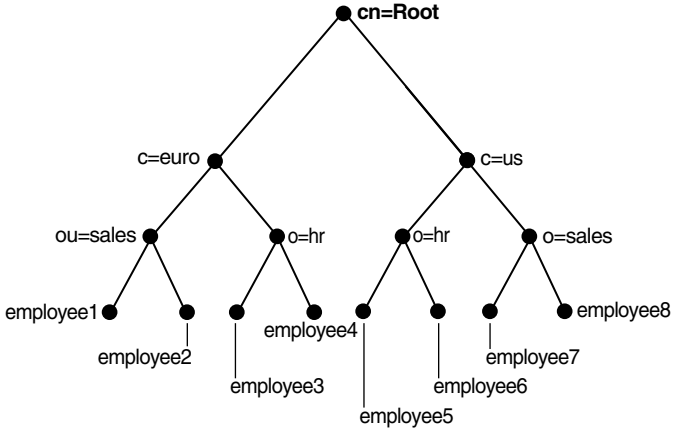
Rules for Creating Attribute Uniqueness

This section describes and gives examples of rules you follow when creating attribute uniqueness constraints. It contains these topics:

- Specifying Multiple Attribute Names in an Attribute Uniqueness Constraint
- Specifying Multiple Subtrees in an Attribute Uniqueness Constraint
- Specifying Multiple Scopes in an Attribute Uniqueness Constraint
- Specifying Multiple Object Classes in an Attribute Uniqueness Constraint
- Specifying Multiple Subtrees, Scopes, and Object Classes in an Attribute Uniqueness Constraint

To understand the examples in this section, refer to Figure 8–1.

Figure 8–1 Example of a Directory Information Tree



Specifying Multiple Attribute Names in an Attribute Uniqueness Constraint

When multiple attribute uniqueness constraints have different values in `orcluniqueattrname`, their effects are independent of each other.

For example, suppose that a user defines two attribute uniqueness constraints as follows:

Constraint1:

```
orcluniqueattrname: employee_id
```

Constraint2:

```
orcluniqueattrname: email_id
```

In this example, Constraint1 and Constraint2 enforce uniqueness on the specified attribute within their own attribute uniqueness scopes. Constraint1 and Constraint2 are independent of each other.

Specifying Multiple Subtrees in an Attribute Uniqueness Constraint

When multiple attribute uniqueness constraints have the same values in `orcluniqueattrname`, `orcluniquescope` and `orcluniqueobjectclass`, but

different values in orcluniquesubtree, the union of subtree scopes specified by those attribute uniqueness constraints is checked.

For example, refer to Figure 8–1 on page 8-4. Suppose that a user defines two attribute uniqueness constraints as follows:

Constraint1:

```
orcluniqueattrname: employee_id
orcluniquesubtree: o=sales, c=us, cn=root
orcluniquescope: onelevel
```

Constraint2:

```
orcluniqueattrname: employee_id
orcluniquesubtree: o=hr, c=euro, cn=root
orcluniquescope: onelevel
```

In this example, the attribute uniqueness on employee_id is enforced against all entries under subtree o=sales,c=us,cn=root and o=hr,c=euro,cn=root—that is, the directory server enforces the unique value of the employee_id attribute for employee1, employee2, employee5 and employee6.

Specifying Multiple Scopes in an Attribute Uniqueness Constraint

When multiple attribute uniqueness constraints have the same values in orcluniqueattrname, orcluniquesubtree and orcluniqueobjectclass, but different values in orcluniquescope, the attribute uniqueness constraint with the largest search scope takes effect.

For example, referring to Figure 8–1 on page 8-4, suppose that a user defines two attribute uniqueness constraints as follows:

Constraint1:

```
orcluniqueattrname: employee_id
orcluniquesubtree: c=us, cn=root
orcluniquescope: onelevel
```

Constraint2:

```
orcluniqueattrname: employee_id
orcluniquesubtree: c=us, cn=root
orcluniquescope: sub
```

In this example, the attribute uniqueness on `employee_id` is enforced against all entries under the subtree `c=us, cn=root` and the entry `c=us, cn=root` itself. Note that this is the same as if the user had defined only Constraint2.

Specifying Multiple Object Classes in an Attribute Uniqueness Constraint

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.

For example, referring to Figure 8–1 on page 8-4, suppose that a user defines two attribute uniqueness constraints as follows:

Constraint1:

```
orcluniqueattrname: employee_id
orcluniquesubtree: c=us, cn=root
orcluniqueobjectclass: person
```

Constraint2:

```
orcluniqueattrname: employee_id
orcluniquesubtree: c=us, cn=root
```

In this example, the attribute uniqueness on `employee_id` is enforced against all entries under the subtree `c=us, cn=root` and the entry `c=us, cn=root` itself, no matter what object class those entries belong to. Note that Constraint2 specifies no `orcluniqueobjectclass` attribute, which is the same as specifying all object classes.

Specifying Multiple Subtrees, Scopes, and Object Classes in an Attribute Uniqueness Constraint

When multiple attribute uniqueness constraints have the same values in `orcluniqueattrname`, but different values in `orcluniquesubtree`, `orcluniquescope`, and `orcluniqueobjectclass`, the union of entries that belong to the attribute uniqueness scopes of different constraints are checked.

For example, referring to Figure 8–1 on page 8-4, suppose that a user defines two attribute uniqueness constraints as follows:

Constraint1:

```
orcluniqueattrname: employee_id
```

```
orcluniquesubtree: o=sales, c=us, cn=root
orcluniquescope: onelevel
orcluniqueobjectclass: person
```

Constraint2:

```
orcluniqueattrname: employee_id
orcluniquesubtree: c=euro, cn=root
orcluniquescope: sub
orcluniqueobjectclass: organization
```

In this example, the attribute uniqueness on employee_id is enforced against the following:

- All entries under the subtree `o=sales,c=us,cn=root` where their object class belongs to person
- All entries under subtree `c=euro,cn=root` and the entry `c=euro,cn=root` itself where their object class belongs to `organization`

Managing Attribute Uniqueness

This section contains these topics:

Location of Attribute Uniqueness Entries

Attribute uniqueness constraint entries are stored under `cn=unique,cn=Common,cn=Products,cn=OracleContext`.

Managing Attribute Uniqueness by Using Oracle Directory Manager

You can use Oracle Directory Manager to create, modify, and delete attribute uniqueness constraint entries.

Creating an Attribute Uniqueness Constraint Entry

1. In the navigator pane, expand in succession Oracle Internet Directory **Servers**, ***directory server instance***, and **Attribute Uniqueness Management**. The

Attribute Uniqueness Management window displays a list of existing attribute uniqueness constraint entries in the right pane.

2. On the toolbar, choose **Create**. This displays the New Constraint window.

   In the New Constraint dialog box, enter values for the fields. These are described in Table C–6 on page C-4.

3. Choose **OK**. This returns you to the Attribute Uniqueness Management window. The entry you just created appears in the list of attribute uniqueness constraint entries.

4. Choose **Apply**.

Modifying an Attribute Uniqueness Constraint Entry by Using Oracle Directory Manager

To modify an attribute uniqueness constraint entry:

1. In the navigator pane, expand in succession Oracle Internet Directory **Servers**, ***directory server instance***, and **Attribute Uniqueness Management**. The Attribute Uniqueness Management window displays a list of existing attribute uniqueness constraint entries in the right pane.

2. In the Attribute Uniqueness Management window, select the attribute uniqueness constraint entry you want to modify, then choose **Edit**. The Attribute Uniqueness Constraint window for that attribute appears.

3. In the Attribute Uniqueness Constraint window, enter your modifications in the appropriate fields, then choose **OK**. This returns you to the Attribute Uniqueness Management window.

4. Choose **Apply**.

Deleting an Attribute Uniqueness Constraint Policy by Using Oracle Directory Manager

To delete an attribute uniqueness constraint policy:

1. In the navigator pane, expand in succession Oracle Internet Directory **Servers**, ***directory server instance***, and **Attribute Uniqueness Management**. The Attribute Uniqueness Management window displays a list of existing attribute uniqueness constraint entries in the right pane.

2. In the Attribute Uniqueness Management window, select the attribute uniqueness constraint entry you want to delete, then choose **Edit**. The Attribute Uniqueness Constraint window for this attribute appears.

3. Choose **Delete**, then, when prompted, confirm the deletion. This returns you to the Attribute Uniqueness Constraint window. The entry you deleted no longer appears in the list of attribute uniqueness constraint entries.

Managing Attribute Uniqueness by Using Command-Line Tools

This section contains these topics:

Enabling and Disabling Attribute Uniqueness by Using Command-Line Tools

You can enable or disable attribute uniqueness for an existing attribute uniqueness constraint entry.

To enable attribute uniqueness for an existing attribute uniqueness constraint entry:

1. Set the `orcluniqueenable` attribute to `1` by using ldapmodify.
2. Restart the directory server to enable the policy.

To disable attribute uniqueness:

1. Set the `orcluniqueenable` attribute to `0` by using ldapmodify.
2. Restart the directory server to disable the policy.

Creating Attribute Uniqueness Constraint Entries by Using Command-Line Tools

To enable attribute uniqueness, specify an attribute uniqueness constraint entry with the attributes listed in Table 8–1 on page 8-2.

Creating Attribute Uniqueness Across an Entire Directory by Using Command-Line Tools To create an instance of attribute uniqueness across an entire directory, specify an attribute name for which you want to enforce value uniqueness.

For example, to make employee identifiers unique for all US employees at MyCompany, you would follow these steps.

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=MyCompany, c=US
orcleuniqueobjectclass: person
```

2. To apply the attribute uniqueness feature, load the attribute uniqueness constraint entry as follows:

```
ldapadd -h host -p port -D DN -w password -f constraint1.dat
```

3. Restart the directory server.

Creating Attribute Uniqueness Across One Subtree by Using Command-Line Tools To create an instance of attribute uniqueness across one or more subtrees, specify:

- An attribute name for which you want to enforce value uniqueness
- Subtree locations under which you want the uniqueness constraint to be enforced

For example, suppose that MyCompany hosts the directories for SubscriberCompany1 and SubscriberCompany2, and you want to enforce the uniqueness of the employee identifier attribute in SubscriberCompany1 only. When you add an entry such as `uid=dlin,ou=people,o=SubscriberCompany1,dc=MyCompany,dc=com`, you must enforce uniqueness only in the `o=SubscriberCompany1,dc=MyCompany,dc=com` subtree. Do this by listing the DN of the subtree explicitly in the attribute uniqueness constraint configuration.

In this case, the LDIF file would look like this:

```
dn: cn=constraint1, cn=unique, cn=common, cn=products, cn=oraclecontext
objectclass: orclUniqueConfig
orcluniqueattrname: employeenumber
orcluniquesubtree: o=SubscriberCompany1,dc=MyCompany,dc=com
```

Creating Attribute Uniqueness Across One Object Class by Using Command-Line Tools To create an instance of attribute uniqueness across one object class, specify:

- An attribute name for which you want to enforce value uniqueness
- Object class name

In this case, the LDIF file would look like this:

```
dn: cn=constraint1, cn=unique, cn=common, cn=products, cn=oraclecontext
objectclass: orclUniqueConfig
orcluniqueattrname: employeenumber
orcleuniqueobjectclass: person
```

Modifying Attribute Uniqueness Constraint Entries by Using Command-Line Tools

To modify an attribute uniqueness entry, use create an LDIF file for the entry, then use ldapmodify to upload it into the directory.

For example, suppose there is an existing attribute uniqueness constraint entry:

```
dn: cn=constraint1, cn=unique, cn=common, cn=products, cn=oraclecontext
objectclass: orclUniqueConfig
orcluniqueattrname: employeenumber
orcluniquesubtree: o=MyCompany, c=US
orcleuniqueobjectclass: person
```

To enforce the constraint against `c=US`, instead of `o=MyCompany`, you would perform these steps:

1. Create an LDIF entry to change the `orcluniquenesssubtree`:

   ```
   dn: cn=constraint1, cn=unique, cn=common, cn=products, cn=oraclecontext
   changetype: modify
   replace: orcluniquesubtree
   orcluniquesubtree: o=Oracle Corporation, c=US
   ```

2. Use ldapmodify to apply the change to directory server.

   ```
   ldapmodify -p port -D user -w password -f file_name
   ```

3. Restart the directory server to effect this change.

Deleting Attribute Uniqueness Constraint Entries by Using Command-Line Tools

Use the `ldapdelete` command-line tool to delete an attribute uniqueness constraint policy.

1. Remove the attribute uniqueness constraint entry from the directory by using ldapdelete.

   ```
   ldapdelete -p port -D bind_DN -w password
   "cn=constraint1,cn=unique,cn=common,cn=products,cn=oraclecontext"
   ```

2. Restart the directory server to effect this change.

Limitations of Attribute Uniqueness in Oracle Internet Directory 10*g* (9.0.4)

When an attribute uniqueness constraint is present in the Oracle Internet Directory replication environment, be careful about configuring the attribute uniqueness constraints on each server. This section contains these topics:

Simple Replication Scenario

Because all modifications by client applications are performed on the supplier server, the attribute uniqueness constraint should be enabled on that 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 directory server from operating correctly, but it can cause a performance degradation.

Multimaster Replication Scenario

In a multimaster replication scenario, nodes serve as both suppliers and consumers of the same replica. Multimaster 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. However, when the changes are later replicated to the other node, the conflict becomes apparent. You must take this type of conflict resolution into consideration as well, deciding whether conflict resolution should be the replication server's responsibility.

9

Dynamic and Static Groups in Oracle Internet Directory

This chapter explains how to administer both static and dynamic groups in Oracle Internet Directory. This chapter contains these topics:

About Groups

Oracle Internet Directory enables you to assign and manage membership in two types of groups—namely, static groups and dynamic groups. Each type of group suited for a different purpose.

This section contains these topics:

Static Groups

A static group is one whose entry contains a list of members that you explicitly administer.

A static group requires you to explicitly administer its membership. For example, if a member changes his name, then you need to change that user's DN for each group he belongs to. For this reason, a static group is best suited for a group whose membership is unlikely to change frequently. Moreover, because a static group contains a list of member DNs, its footprint in the directory increases with the membership list. For this reason, it is best suited for a group whose entries take up relatively less space in the directory.

Schema Elements for Creating Static Groups

When you create the entry for this kind of group, you associate it with either the `groupOfNames` or `groupOfUniqueNames` object class.

Each of these object classes has a multivalued attribute for storing the names of group members. To assign a user as a member of a group, you add the DN of each member to the respective multivalued attribute. Conversely, to remove a member from a group, you delete the member's DN from the respective attribute. In the `groupOfNames` object class, this multivalued attribute is `member`, and, in the `groupOfUniqueNames` object class, it is `uniqueMember`.

Dynamic Groups

A dynamic group is one whose membership, rather than being maintained in a list, is computed on the fly, based on rules and assertions you specify. For example, suppose that you want to send an e-mail to all users in the `ou=americas` naming context. To do this, you create a dynamic group in which you specify `ou=americas` as the naming context of interest. You further specify that you want only e-mail addresses returned. When the e-mail application queries the directory for that particular group, the directory server computes the membership dynamically and returns the corresponding list of e-mail addresses.

To use another example, suppose you want to send an e-mail to all employees who report to a manager named Anne Smith. In this case, you do not specify a naming context, as in the previous example. Instead, you create a dynamic group specifying that you want to retrieve the e-mail addresses of all employees reporting to Anne Smith. As in the previous example, when the e-mail application queries the directory for that particular group, the directory server computes the membership dynamically and returns the corresponding list of e-mail addresses.

Note: in this example, the e-mail application specifies that the directory server is to read the specific attributes of the members—rather than the membership lists. It does this by passing the control 2.16.840.1.113894.1.8.5.

Also, when querying for the groups that a user belongs to, the application can direct that dynamic groups, in addition to static groups, be queried. For this to happen, it passes the control 2.16.840.1.113894.1.8.7. If this control is not passed, then only static groups are queried.

Schema Elements for Creating a Dynamic Group

When you create a dynamic group, you begin as when creating a static group—that is, you associate its entry with either the `groupOfNames` or `groupOfUniqueNames` object class. You then associate that object class with the auxiliary object class `orclDynamicGroup`. This auxiliary object class has various attributes in which you specify one of two methods for dynamically computing the membership of the group.

The two methods are:

- Using the `labeledURI` attribute

When using this method, the directory server performs a typical search based on the hierarchy of the DIT. It requires you to provide a value for one of the attributes of the `orclDynamicGroup` object class, namely `labeledURI`. In this attribute, you specify the base of the query, the filters, and any required attributes. For example, suppose that you have entered the following value for the `labeledURI` attribute:

```
labeledURI:ldap://host name/"ou=MyOrganizationalUnit,o=MyCompany,c=US"??sub"
(objectclass=person)
```

When you use this method, a search for the entry returns entries for all members of the group.

> **See Also:** *The LDAP URL Format* (Request for Comments 2255). T. Howes, M. Smith, December 1997; available on the World Wide Web at `http://www.ietf.org` for more information about how LDAP URLs are to be represented—as, for example, in the `labeledURI` attribute.

- Using a `CONNECT BY` assertion

  Unlike the previous method, this method relies not on the hierarchy of the DIT, but on attributes that implicitly connect entries to each other, regardless of their location in the DIT. For example, the `manager` attribute connects the entries of employees with those of their managers, and this connection applies regardless of the location of the employee entries in the DIT. This method uses a `CONNECT BY` clause in which you specify the attribute to use for building the hierarchy—for example, Manager—and the starting value for such a hierarchy—for example, `cn=Anne Smith`.

  More specifically, to use this method, you specify in the `orclDynamicGroup` object class a value for each of the single-valued attributes in Table 9–1.

Table 9–1 orclDynamicGroup Attributes for "Connect By" Assertions

Attribute	Description
`orclConnectByAttribute`	The attribute that you want to use as the filter for the query—for example, `manager`
`orclConnectByStartingValue`	The DN of the attribute you specified in the `orclConnectByAttribute` attribute—for example, Anne Smith

For example, to retrieve the entries of all employees who report to Anne Smith in the MyOrganizational Unit in the Americas, you would provide values for these attributes as follows:

```
orclConnectByAttribute=manager
orclConnectByStartingValue=
"cn=Anne Smith,ou=MyOrganizationalUnit,o=MyCompany,c=US"
```

You can also develop an application specifying that you want the values for a particular attribute—for example, the `email` attribute—of all the members.

> **See Also:** *Oracle Internet Directory Application Developer's Guide* for more information about how to develop applications that retrieve values for particular attributes

Hierarchies

Hierarchies can be either explicit or implicit.

In explicit hierarchies, the relationship is determined by the location of the entry in the DIT—for example, Group A may reside higher in the DIT than Group B.

In implicit hierarchies, the relationship between entries is determined not by the location in the DIT, but by the values of certain attributes. For example, suppose that you have a DIT in which the entry for John Doe is at the same level of the hierarchy as Anne Smith. However, suppose that, in the entry for John Doe, the `manager` attribute specifies Anne Smith as his manager. In this case, although their locations in the DIT are at an equal level, their rankings in the hierarchy are unequal because Anne Smith is specified as John Doe's manager.

Note: If you create a hierarchical group, be sure that it is truly hierarchical. For example, in a true hierarchy, Group A can be a member of Group B, but Group B cannot at the same time be a member of Group A. Because the latter relationship is cyclical, a search for the members of Group A fails.

In a query based on an implicit hierarchy, the client can specify in the search request the control 2.16.840.1.113894.1.8.3. The filter in this query specifies the attribute used to build the implicit hierarchy. For example, `(manager=cn=john doe, o=foo)` specifies the query for all people reporting directly or indirectly to John Doe. The implicit hierarchy is based on the `manager` attribute. The base of the search is ignored for such queries.

Querying Group Entries

An application can query either kind of group to do the following:

- List all members of a group
- List all groups of which a user is a member
- Check to see if a user is a member of a particular group

In addition, you can query dynamic groups, but not static ones, for whatever member attributes you specify.

When to Use Each Kind of Group

When deliberating about which kind of group to use, you need to weigh the ease of administration against higher performance. For example, dynamic groups provide for easier administration, but cause a decrease in performance. Table 9–2 lists some things to consider when deliberating whether to use static or dynamic groups.

Table 9–2 Static and Dynamic Group Considerations

Consideration	Static Groups	Dynamic Groups
Ease of administration	More difficult to administer if group memberships are large and change frequently	Easier to use, especially when group memberships are large and change frequently
Performance	Higher level of performance because you explicitly administer the membership list	Decreased level of performance because memberships are computed on the fly
Size of footprint in the directory	Larger footprint depending on the size of group memberships	Small footprint regardless of size of group memberships

Limitations of Dynamic Groups in Oracle Internet Directory 10*g* (9.0.4)

This version of Oracle Internet Directory does not support the use of dynamic groups in access control lists. You cannot associate dynamic groups with either the `orclACPgroup` or the `orclPrivilegeGroup` object class.

When querying dynamic group for required attributes of the member, this release supports reading the attributes only of members not explicitly listed in the membership list. Also, in this case, an ldapsearch filter based on membership—that is, `member` or `uniqueMember`—cannot be applied to the dynamic group object.

The hierarchical group resolution query works only for static groups. If a dynamic groups is a member of a static group, then the query to resolve the entire hierarchy of the groups does not evaluate the dynamic groups. Thus, if a static Group A is a member of another static Group B which in-turn is a member of static Group C, then the query to compute all the groups that a user is a member of (assuming the user is a member of static Group A) correctly returns groups A, B, and C. However, if group C is a dynamic group, then the same query returns only Groups A and B.

The `CONNECT BY` query to resolve implicit hierarchies works only with the equality filter. The base of the search is not used while executing this kind of query.

Managing Group Entries

This section contains these topics:

- Managing Static Group Entries by Using Oracle Directory Manager
- Managing Static Group Entries by Using Command-Line Tools
- Managing Dynamic Groups by Using Command-Line Tools

Note: If you are creating a hierarchy of groups, be sure that it is a true hierarchy as described in "Hierarchies" on page 9-5.

See Also:

- "Security Groups" on page 14-3 for instructions on setting access control policies for group entries
- Globalization Support on page 2-13 and Chapter 14, "Directory Access Control" for information about access privileges

Managing Static Group Entries by Using Oracle Directory Manager

You can use Oracle Directory Manager to both create and modify static group entries.

Creating Static Group Entries by Using Oracle Directory Manager

If the entry belongs to the `groupOfNames` object class, then you determine membership in the group by adding DNs to the multivalued attribute `member`. If the entry belongs to the `groupOfUniqueNames` object class, then you determine

membership in the group by adding DNs to the multivalued attribute `uniqueMember`.

To add a static group entry:

1. Expand in succession **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Entry Management**.
3. On the toolbar, choose **Create**. The New Entry dialog box appears.
4. In the **Distinguished Name** field, type the full DN. You may also use **Browse** 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.
5. To specify the object classes you want to use for the new entry, to the right of the **Object Classes** box, choose **Add**. The Super Class Selector dialog box appears.
   a. In the Super Class Selector dialog box, select the following object classes:
      * `top`
      * Either `groupOfNames` or `groupOfUniqueNames`
   b. Choose **Select**. The object classes you selected appear in the **Object Classes** window of the New Entry dialog box.
6. 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. Choose **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 choose **OK**. This returns you to the New Entry dialog box. The entry you just selected is added to the list in the members window.
7. Choose **OK**.

Modifying a Static Group Entry by Using Oracle Directory Manager

To modify the member list for a group entry:

1. Perform a search for the group entry you want to modify.
2. In the right pane, in the **Distinguished Name** box, select the group entry you want to modify.
3. Choose **Edit**.
4. In the Entry dialog box, scroll to the text area for the `member` attribute and modify the value.
5. Choose **OK**.

Managing Static Group Entries by Using Command-Line Tools

This section provides examples of how you create and modify static group entries.

Creating a Static Group Entry by Using ldapadd

The syntax for the LDIF file is:

```
dn: DN_of_group_entry
objectclass: top
objectclass: [groupOfNames] [groupOfUniqueNames]
member: DN of member 1
member: DN of member 2
.
.
.
member: DN of member N
```

The following command adds this LDIF file to the directory:

```
ldapadd -p port_number -h host -f file_name.ldif
```

Example: Creating a Static Group Entry by Using ldapadd The following example shows an LDIF file named `myStaticGroup.ldif` for the entry for a group named MyStaticGroup:

```
dn: cn=myStaticGroup,c=us
objectclass: top
objectclass: groupOfNames
member: cn=John Doe
member: cn=Anne Smith
```

The following command adds this LDIF file to the directory:

```
ldapadd -p 389 -h myhost -f myStaticGroup.ldif
```

Modifying a Static Group by Using ldapmodify

To add a member to a group, the syntax of the LDIF file is:

```
dn: DN_of_group_entry
changetype: modify
add:member
member:DN of member entry
```

To delete a member from a group, the syntax of the LDIF file is:

```
dn: DN_of_group_entry
changetype: modify
delete:member
member:DN of member entry
```

Issue this command to modify the file:

```
ldapmodify -p 389 -v  -f file_name.ldif
```

where -v specifies verbose mode.

Example: Modifying a Static Group by Using ldapmodify The following example adds John Doe to a group named MyStaticGroup. As in the previous example, the data for this user entry is in the `myStaticGroup.ldif` file. This file contains the following:

```
dn: cn=myStaticGroup,c=us
changetype: modify
add:member
member: John Doe
```

Issue this command to modify the file:

```
ldapmodify -p 389 -v  -f myStaticGroup.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 Dynamic Groups by Using Oracle Directory Manager

You can use Oracle Directory Manager to both create and modify static group entries.

Creating Dynamic Group Entries by Using Oracle Directory Manager

If the entry belongs to the `groupOfNames` object class, then you determine membership in the group by adding DNs to the multivalued attribute `member`. If the entry belongs to the `groupOfUniqueNames` object class, then you determine membership in the group by adding DNs to the multivalued attribute `uniqueMember`.

To add a dynamic group entry:

1. Expand in succession **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Entry Management**.
3. On the toolbar, choose **Create**. The New Entry dialog box appears.
4. In the **Distinguished Name** field, type the full DN. You may also use **Browse** 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.
5. To specify the object classes you want to use for the new entry, to the right of the **Object Classes** box, choose **Add**. The Super Class Selector dialog box appears.
   a. In the Super Class Selector dialog box, select the following object classes:
      * `top`
      * `orcldynamicgroup`
      * Either `groupOfNames` or `groupOfUniqueNames`
   b. Choose **Select**. The object classes you selected appear in the **Object Classes** window of the New Entry dialog box.
6. Enter the mandatory and optional attributes for your group entries.

   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. Choose **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 choose **OK**. This returns you to the New Entry dialog box. The entry you just selected is added to the list in the members window.

7. Choose **OK**.

Modifying a Dynamic Group Entry by Using Oracle Directory Manager

To modify the member list for a dynamic group entry:

1. Perform a search for the group entry you want to modify.
2. In the right pane, in the **Distinguished Name** box, select the group entry you want to modify.
3. Choose **Edit**.
4. In the Entry dialog box, scroll to the text area for the `member` attribute and modify the value.
5. Choose **OK**.

Managing Dynamic Groups by Using Command-Line Tools

This section tells you how to create and modify dynamic groups by using command-line tools.

Creating a Dynamic Group Entry by Using ldapadd

If you use the `labeledURI` attribute, then the syntax for the LDIF file is:

```
dn: DN_of_group_entry
objectclass: top
objectclass: [groupOfNames] [groupOfUniqueNames]
objectclass: orcldynamicgroup
labeledURI:ldap:ldap_URL
member: DN of member 1
member: DN of member 2
.
.
.
member: DN of member N
```

The following command adds this LDIF file to the directory:

```
ldapadd -p port_number -h host -f file_name.ldif
```

If you use the `CONNECT BY` string, then the syntax for the LDIF file is:

```
dn: DN_of_group_entry
objectclass: top
objectclass: [groupOfNames] [groupOfUniqueNames]
objectclass: orcldynamicgroup
orclConnectByAttribute=manager
orclConnectByStartingValue=
"cn=Anne Smith,ou=MyOrganizationalUnit,o=MyCompany,c=US"
member: DN of member 1
member: DN of member 2
.
.
.
member: DN of member N
```

Example: Creating a Dynamic Group Entry by Using ldapadd

The following example shows an LDIF file for the entry for a dynamic group:

```
dn: cn=myDynamicGroup,c=us
objectclass: top
objectclass: groupOfNames
objectclass: orcldynamicgroup
labeledURI:ldap:
//my_host/ou=MyNeworganizationalUnit,o=MyCompany,c=US??sub?(objectclass=person)
member: cn=John Doe
member: cn=Anne Smith
```

The following command adds this LDIF file to the directory:

```
ldapadd -p 389 -h myhost -f myDynamicGroup.ldif
```

Example: Modifying a Dynamic Group by Using ldapmodify

To change the organizational unit of the group created in the previous example, the syntax of the LDIF file is:

```
dn: DN_of_group_entry
changetype: modify
replace:labeledURI
labeledURI:ldap:
//my_host/ou=MyNeworganizationalUnit,o=MyCompany,c=US??sub?(objectclass=person)
```

Note: When you add or modify an entry, the Oracle directory server does not verify the syntax of the attribute values in the entry.

10

Logging, Auditing, and Monitoring the Directory

Oracle Internet Directory provides a comprehensive framework for enabling you to debug, audit, and monitor the directory. This chapter contains these topics:

- Using Debug Logging
- Using the Audit Log
- Monitoring Oracle Internet Directory Servers

Using Debug Logging

This section contains these topics:

About Oracle Internet Directory Debug Logging

Oracle Internet Directory enables you to:

- View logging information for the directory server, the directory replication server, and the directory integration server
- Set the logging level
- Specify one or more operations for which you want logging to occur
- Search messages in a standard format to determine remedial action for fatal and serious errors
- View trace messages according to their severity and order of importance
- Diagnose Oracle Internet Directory components by examining trace messages with relevant information about, for example, entry DN, ACP evaluation, and the context of an operation

About Log Messages

This section discusses log messages—those associated with specified LDAP operations and those not. It provides an example of a trace log and explains how to interpret it.

Log Messages for Specified LDAP Operations

Log messages for a specified operation are stored as a trace object. This object tracks the operation from start to finish across the various Oracle Internet Directory modules. It is entered in the log file when one of the following occur:

- An LDAP operation completes

- A high priority message is logged
- The trace messages buffer is full

Each thread has one contiguous block of information for each operation, and that block is clearly delimited. This makes it easy, in a shared server environment, to follow the messages of different threads, operations, and connections.

If, because of an internal message buffer overflow, a single trace object cannot contain all the information about an operation, then the information is distributed among multiple trace objects. Each distributed piece of information is clearly delimited and has a common header. To track the progress of the operation, you follow the trace objects and their common header to the end, which is marked with the trace message "Operation Complete".

Log Messages Not Associated with Specified LDAP Operations

Messages not associated with any LDAP operation are represented in a simple format, which is not object-based. It is entered in the log file when either the operation completes or a high priority message is encountered.

Example: Trace Messages in Oracle Internet Directory Server Log File

```
2003/01/28:13:44:27 * Main:1 * Starting up the OiD Server, on node dthakuri-sun

2003/01/28:13:44:27 * Main:1 * Oid Server Connected to DB store via inst1
connect string.
2003/01/28:13:44:27 * Main:1 * OiD LDAP server started.

2003/01/28:13:44:31 * ServerController:1 * INFO * slsfctSpawnDispatcher * Entry
2003/01/28:13:44:31 * ServerController:1 * INFO * gslsfctSpawnDispatcher *
Spawned server dispatcher thread successfully. Thread id : 1
2003/01/28:13:44:31 * ServerController:1 * INFO * gslsfctSpawnDispatcher * Exit


2003/01/28:13:44:31 * ServerWorker:6 * INFO : ServerWorker : Entry
2003/01/28:13:44:31 * ServerWorker:6 * INFO : gslsfccRegisterThread : Entry
2003/01/28:13:44:31 * ServerWorker:6 * INFO : gslsfccRegisterThread : Exit
2003/01/28:13:44:31 * ServerWorker:6 * INFO * gslfsfAStr2Filter *
Filter="(|(objectclass=referral))"
2003/01/28:13:44:31 * ServerWorker:6 * INFO * gslfsfAStr2Filter *
Filter="(objectclass=referral)"
2003/01/28:13:44:31 * ServerWorker:6 * INFO * gslfsfCStr2Simple *
Filter="objectclass=referral"
2003/01/28:13:44:31 * ServerWorker:6 * INFO * gslsbnrNormalizeString() String to
```

```
Normalize: "objectclass"
2003/01/28:13:44:31 * ServerWorker:6 * INFO * gslsbnrNormalizeString()
Normalized value: "objectclass"


BEGIN
2003/01/28:13:45:49 * ServerWorker:6 * ConnID:0 * OpId:0 * OpName:bind
13:45:49 * INFO * gslfbiADoBind * Entry
13:45:49 * INFO * gslfbiGetControlInfo * Entry
13:45:49 * INFO * gslfbiGetControlInfo * Exit
13:45:49 * INFO * gslfbiADoBind * connID=0 opID=0 Version=3 BIND dn=""
method=128
13:45:49 * INFO * gslfrsBSendLdapResult * Entry
13:45:49 * INFO * gslfrsASendLdapResult2 * Entry
13:45:49 * INFO * sgslunwWrite * Entry
13:45:49 * INFO * sgslunwWrite * Exit
13:45:49 * INFO * gslfrsASendLdapResult2 * Exit
13:45:49 * INFO * gslfrsBSendLdapResult * Exit
13:45:49 * INFO * gslfbiADoBind * Exit
13:45:49 * INFO * Total Bind operation time for dn=2588  micro sec and Total
Worker time=3434  micro sec
END

2003/01/28:13:45:49 * ServerWorker:6 * INFO * ServerWorker * Operation Complete

2003/01/28:13:44:31 * ServerWorker:7 * INFO * ServerWorker : Entry
2003/01/28:13:44:31 * ServerWorker:7 * INFO * gslsfccRegisterThread : Entry
2003/01/28:13:44:31 * ServerWorker:7 * INFO * gslsfccRegisterThread : Exit


BEGIN
2003/01/28:13:48:53 * ServerWorker:13 * ConnID:0 * OpId:0 * OpName:bind
13:48:14 * INFO * gslfbiADoBind * Entry
13:48:53 * INFO * gslfbiGetControlInfo * Entry
13:48:53 * INFO * gslfbiGetControlInfo * Exit
13:48:53 * INFO * gslfbiADoBind * conn=0 op=0 Version=3 BIND dn="cn=proxy"
method=128
13:48:53 * INFO * gslsbbBind * Entry
13:48:53 * INFO * gslsbnrNormalizeString * String to Normalize: "proxy"
13:48:53 * INFO * gslsbnrNormalizeString * Normalized value: "proxy"
13:48:53 * INFO * gslfrsBSendLdapResult * Entry
13:48:53 * INFO * gslfrsASendLdapResult2 * Entry
13:48:53 * INFO * sgslunwWrite * Entry
13:48:53 * INFO * sgslunwWrite * Exit
13:48:53 * INFO * gslfrsASendLdapResult2 * Exit
```

```
13:48:53 * INFO * gslfrsBSendLdapResult * Exit
13:48:53 * INFO * gslsbbBind * Exit
13:48:53 * INFO * gslfbiADoBind:Exit
13:48:53 * INFO * Total Bind operation time for dn = cn=proxy is  3710  micro
sec
 Total Worker time = 4767  micro sec
END

2003/01/28:13:48:53 * ServerWorker:13 * INFO * ServerWorker * Operation Complete


2003/01/28:14:05:56 * ServerWorker:6 * FATAL * ServerWorker * Processing
shutdown notification
2003/01/28:14:05:56 * ServerWorker:6 * WARNING * ServerWorker * Shutting down
worker ID :  6
```

How to Interpret Trace Messages in the Log File

As shown in the sample messages in the previous section, log information can be associated with either a thread that performs an operation or one that does not. In the case of a thread that performs an operation, the header of the log contains:

- Date and time
- Thread name and identifier for the particular connection
- Connection identifier
- The name and identifier of the associated operation

A thread that does not perform an operation logs normal trace messages. Its header contains the date, time, and the thread identifier. It does not contain connection and operation-related information.

A trace object starts with the keyword `BEGIN` and ends with the keyword `END`.

Table 10–1 describes each field in a trace message.

Table 10–1 Fields in Trace Messages

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
For messages not based on objects: Date and time For messages based on objects: Time only	For non-object-based trace messages only, the thread identifier	Trace message criticality. This has four possible values: ▪ FATAL ▪ ERROR ▪ WARN (Warning) ▪ INFO (Informational)	Function name	Information about the operation performed. This information can be used to diagnose problems.	Error code, if available. The error code could be for the operating system, the Oracle database, or LDAP.

Setting Debug Logging Levels

You can set debug logging levels by using either **Oracle Directory Manager** or 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.
3. Select **Debug Flags**.

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

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 10–2.

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 10–2 provides the complete list of debug logging levels.

Table 10–2 Debug Logging Levels

Logging Level Value	Provides Information Regarding
1	Heavy trace debugging
128	Debug packet handling
256	Connection management, related to network activities
512	Search filter processing
1024	Entry parsing
2048	Configuration file processing
8192	Access control list processing
491520	Log of communication with the back end - that is with the database
524288	Schema related operations
4194304	Replication specific operations
8388608	Log of entries, operations and results for each connection
16777216	Trace function call arguments
67108863	All possible operations/data

For example, to trace search filter processing (512) and active connection management (256), enter 768 as the debug level (512 + 256 = 768) as follows:

```
oidctl server=oidldapd instance=1 flags='-debug 768' restart
oidctl server=oidrepld instance=1 flags='-h my_host -p 389 -d 768' restart
```

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

Setting the Operation Debug Dimension

To make logging more focused, use the debug dimensions in conjunction with the debug levels. For example, to limit logging to particular directory server operations, specify the debug dimension to those operations.

Table 10–3 shows these dimensions.

Table 10–3 Debug Dimension Values for LDAP Operations

Operation Debug Dimension Value	Provides Information Regarding
1	ldapbind
2	ldapunbind
4	ldapadd
8	ldapdelete
16	ldapmodify
32	ldapmodrdn
64	ldapcompare
128	ldapsearch
256	ldapabandon
511	All LDAP operations

You can set the debug operation dimension by using either Oracle Directory Manager or ldapmodify.

Setting the Operation Debug Dimension by Using Oracle Directory Manager

To set the operation debug dimension:

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.
3. Select **Debug Operation Flag**.

By default, all operations are selected. However, to generate a log for a specific operation, select the corresponding operation. You can select more than one operation.

Setting the Operation Debug Dimension by Using ldapmodify

To log more than one operation, add the values of their dimensions. For example, if you want to trace ldapbind (1), ldapadd (4) and ldapmodify (16) operations, then create an LDIF file setting the `orcldebugop` attribute to 21 (1 + 4 + 16 = 21). The LDIF file is as follows:

```
dn:
changetype:modify
replace:orcldebugop
orcldebugop:21
```

To load this file, enter:

```
ldapmodify -h host_name -p port_number -f file_name
```

Force Flushing the Trace Information to a Log File

To minimize the performance overhead in I/O operations, the debug messages are flushed to the log file periodically instead of every time a message is logged by the directory server. Writing to the log file is performed when one of the following occur:

- An LDAP operation completes
- A high priority message is logged
- The trace messages buffer is full

However, in some situations, you may want to see the trace messages in the log file as they are logged, without having to wait for the periodic flush. To do this, set the DSA configuration attribute `orcldebugforceflush` to `1`. Do this by using ldapmodify as shown in the following example.

To enable force flushing by using ldapmodify, create an LDIF file as follows:

```
dn: cn=dsaconfig,cn=configsets,cn=oracle internet directory
changetype: modify
replace: orcldebugforceflush
orcldebugforceflush: 1
```

To load this file, enter the following:

```
ldapmodify -h host_name -p port_number -f file_name
```

Note:

- When force flushing is enabled, the format of the trace message object for every operation becomes fragmented.
- By default, force flushing is inhibited. After you have flushed the necessary information to the log file, you should disable force flushing.

See Also: Table B–6 on page B-7 for information about the `orcldebugforceflush` attribute

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 10-12 for a listing of audit levels
- "Setting the Audit Level" on page 10-13 for instructions on specifying the audit level
- "Searching for Audit Log Entries by Using Oracle Directory Manager" on page 10-15
- "Searching for Audit Log Entries by Using ldapsearch" on page 10-16
- "ldapdelete Syntax" on page A-28

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:

Table 10–4 Attributes of the orclAuditoc Object Class

Attribute	Description
`orclsequence`	Used to create the name of the entry. The name is generated using a database sequence.
`orcleventtype`	Specifies the type of event that occurred. This is a cataloged attribute.
`orcleventtime`	Specifies the time at which the event occurred. This is formatted in **UTC (Coordinated Universal Time)**. UTC is indicated by a z at the end of the value. For example, `orcleventtime: 199811281010z`
`orcluserdn`	Specifies the identity of the user who logged into the Oracle directory server to perform the operation. This attribute is 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 6-19.

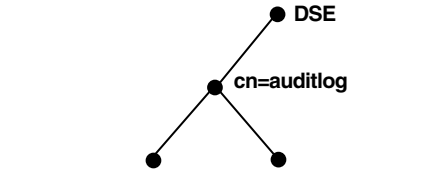
See Also:

- "The Catalog Management Tool (catalog.sh) Syntax" on page A-19 for information about cataloged attributes
- "Object Class Types" on page 2-9 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 10–1.

Figure 10–1 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 10–5 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 10–5 Auditable Events

Event	Description	Audit Levels
Super user 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

Table 10–5 (Cont.) Auditable Events

Event	Description	Audit Levels
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
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. This tab page lists the auditable events described in this table:

Table 10–6 Audit Mask Levels

Audit Level	Description
Super user login	Super user bind to the server (successes or failures)
Schema element add/replace	Addition of a new schema element (successes or failures)
Schema element delete	Deletion of a schema (successes or failures)
Bind	Unsuccessful bind cases
Access violation	Access denied by ACP
DSE modification	Changes to DSE entry (successes or failures)
Replication login	Replication server authentication (successes or failures)
ACL modification	Changes to ACPs
User password modification	Modification of user password attribute
Add	ldapadd operation (successes or failures)
Delete	ldapdelete operation (successes or failures)
Modify	ldapmodify operation (successes or failures)
ModifyDN	ldapModifyDN operation (successes or failures)

3. Select the audit level you want to use.
4. Choose **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 for the changes to take effect.

> **See Also:** "Restarting Oracle Internet Directory Server Instances" on page A-16 for instructions on how to restart the directory server

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 events in Table 10–7 on page 10-15.

Table 10–7 Example: Setting the Audit Level

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 Oracle Internet Directory Server Instances" on page A-16 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 successively Oracle Internet Directory **Servers** and ***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 you want to search for. Because not all attributes are used in every entry, be sure that the attribute you specify actually corresponds to one in the entry that you are searching for. Otherwise, the search fails.
   b. From the list in the middle of the search criteria bar, select a filter. These are described in Table C–37 on page C-35.
   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. These are described in Table C–38 on page C-36.
6. Choose **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 choose to exploit the features of Oracle Internet Directory Server Manageability. The Audit Log Entry dialog box displays the properties for the audit log entry you selected.

> **See Also:** "Configuring the Display and Duration of Searches in Oracle Directory Manager" on page 4-12 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-39

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 connect_string -base "cn=auditlog"
```

Monitoring Oracle Internet Directory Servers

Oracle Internet Directory Server Manageability enables you to monitor various types of information about Oracle Internet Directory servers. This section contains these topics:

Capabilities of Oracle Internet Directory Server Manageability

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. For example, you can view the number of active database sessions over a period of time.
- General statistics about specific server operations—for example, add, modify, or delete operations. For example, you can view the number of directory server operations over a period of time.
- User statistics comprising successful and failed bind and compare operations to the directory and the user performing each one
- Critical events related to system resources and security—for example, occasions when a user provided the wrong password or had inadequate access rights to perform an operation
- Status information of the directory server and the directory replication server—for example, the date and time at which the directory replication server was invoked
- Status information of Oracle directory integration and provisioning server and the integration profiles—for example, the number of times that the directory integration server failed, or whether an integration profile is enabled

See Also: Chapter 32, "Oracle Directory Integration and Provisioning Platform Concepts and Components"

You can view monitored information by using the Oracle Enterprise Manager Application Server Control.

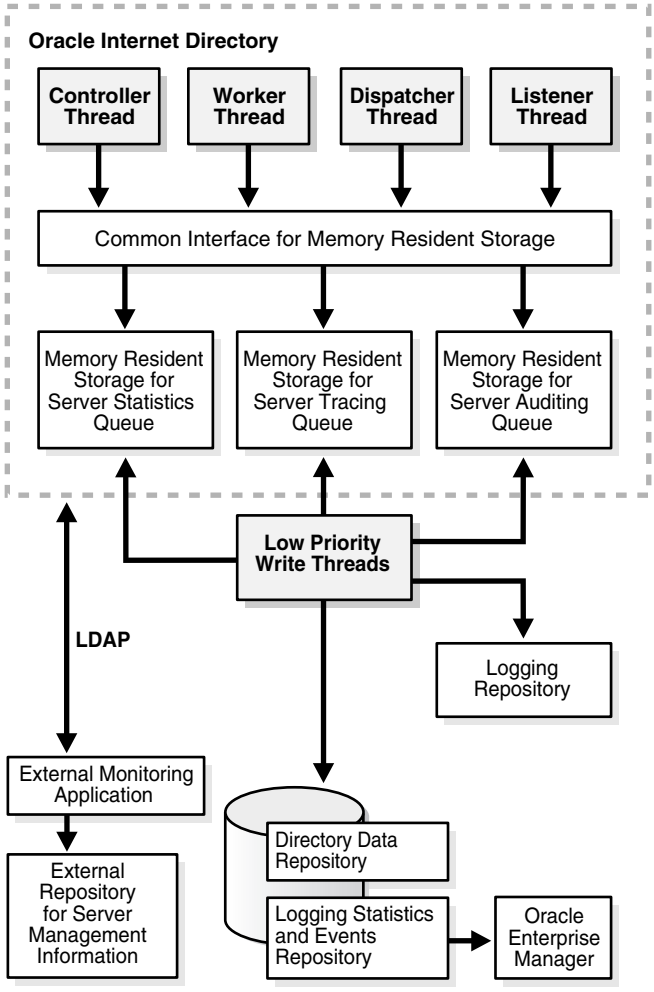
See Also:

- Online help for Oracle Enterprise Manager Application Server Control
- The chapter about administration tools in the *Oracle Application Server 10g Administrator's Guide*

Oracle Internet Directory Server Manageability Architecture and Components

Figure 10–2 and the accompanying text explain the relationship between the various components of directory server manageability.

Figure 10–2 Architecture of Oracle Internet Directory Server Manageability



Oracle Internet Directory A directory server responds to directory requests from clients. It has four kinds of functional threads: controller, worker, dispatcher, and

listener. 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 2-18 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 Application Server Control The Application Server Control 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 for all three modules—namely, server statistics, server tracing, and server auditing—in the DSE root of the directory. To specify periodicity, amount, and level of information to be gathered, you must set appropriate values for these parameters.

Configuring Oracle Internet Directory Server Manageability

To configure the Oracle Internet Directory Server Manageability framework, you use ldapmodify to set positive integer values for various attributes in the root DSE.

- To enable health and general statics, set the `orclStatsFlag` and `orclStatsPeriodicity` attributes.
- To enable user statistics:
  - Set the `orclstatslevel` attribute to `1`
  - Set the `orclStatsPeriodicity` attribute
- To enable critical events, set the `OrclEventLevel` attribute.
- To enable events other than super user, proxy user, and replication administrator login:
  - Set the `OrclEventLevel` attribute to the appropriate value
  - Set the `orclStatsFlag` to `1`

> **See Also:** " Attributes for Oracle Internet Directory Server Manageability" on page B-24 for information about each of the attributes you set when using Oracle Internet Directory Server Manageability

For example, to enable the Oracle Internet Directory Server Manageability framework, you create an LDIF file that looks like this:

```
dn:
changetype: modify
replace: orclstatsflag
orclstatsflag:1
```

To upload this file, enter the following command:

```
ldapmodify -h host -p port_number -D bind_DN -w bind_DN_password -f file_name
```

where the bind DN authorized to perform server manageability configuration is `cn=emd admin,cn=oracle internet directory`.

> **See Also:** Oracle Enterprise Manager Application Server Control online help for more information about monitoring and managing Oracle Internet Directory servers by using Oracle Internet Directory Server Manageability

Configuring Critical Events

To configure critical events, use ldapmodify to set the `OrclEventLevel` attribute to one or more of the event levels listed in Table 10–8.

Table 10–8 Critical Event Levels

Level Value	Critical Event	Information It Provides
1	Super user login	Super uses bind (successes or failures)
2	Proxy user login	Proxy user bind (failures)
4	Replication login	Replication bind (failures)
8	Add access	Add access violation
16	Delete access	Delete access violation
32	Write access	Write access violation
64	ORA 3113 error	ORA-3113 Error

Table 10–8 (Cont.) Critical Event Levels

Level Value	Critical Event	Information It Provides
128	ORA 3114 error	ORA-3114 Error
255	All critical events	

Using the Oracle Internet Directory Server Manageability Framework Through Oracle Enterprise Manager Application Server Control

To exploit the features of Oracle Internet Directory Server Manageability, you use Oracle Enterprise Manager Application Server Control as explained in this section.

Enabling Information Collection by Using Oracle Enterprise Manager Application Server Control

To enable information collection by using Oracle Enterprise Manager Application Server Control:

1. In the Oracle Internet Directory main window, select **LDAP Metrics**. This displays the LDAP Diagnostic Collection Configuration page.
2. Check **Collect Metrics**.
3. Select **Interval**.
4. Enter the required password.
5. Choose **Apply**.

> **Note:** To enable critical events, use ldapmodify to set the attribute `orclEventLevel` to the appropriate value.

Starting a New Directory Server Instance by Using Oracle Enterprise Manager Application Server Control

To start a server:

1. In the Oracle Internet Directory main window, choose **Start New Instance**. The Start a New LDAP Server Instance Window displays a table that enables you to choose a configuration set.

Table 10–9 Fields in the Start a New LDAP Server Instance Window

Column	Description
Set Number	The configuration set number for the directory server instance
Default Port	The default port number for the directory server instance
Port Available	Indicator of whether the default port is available
Maximum Database Connections	The number of database connections this directory instance can accommodate
Server Processes	The number of server processes
Port Number	The port number you assign to the directory server instance if the default port number is not used

2. In the **Set Number** column, select the configuration set you want to use.

   If the default port is not available, then, in the **Port Number** column, specify a port number.

3. Choose **Start**.

Stopping a Directory Server Instance by Using Oracle Enterprise Manager Application Server Control

To stop a directory server instance:

1. In the Oracle Internet Directory main window, in the **LDAP Instances** section, select the directory server instance you want to stop.
2. Choose **Stop**.

Restarting a Directory Server Instance by Using Oracle Enterprise Manager Application Server Control

To restart a directory server instance:

1. In the Oracle Internet Directory main window, in the **LDAP Instances** section, select the server you want to restart.
2. Choose **Restart**. The Restart an LDAP Server Instance window displays the following table.

Table 10–10 Fields in the Restart an LDAP Server Instance Window

Column	Description
Set Number	The configuration set number for the directory server instance
Default Port	The default port number for the directory server instance
Port Available	Indicator of whether the default port is available
Maximum Database Connections	The number of database connections this directory instance can accommodate
Server Processes	The number of server processes
Port Number	The port number you assign to the directory server instance if the default port number is not used

3. Select a configuration. If the default port is not available, then, in the **Port Number** column, enter a port number.
4. Choose **Start**.

Viewing Directory Server Activities by Using Oracle Enterprise Manager Application Server Control

To view directory server activities information:

1. In the Directory Server main window, select the directory server instance whose information you want to view.
2. Choose **View Load**. The LDAP Load window appears.
3. From the **Select Load Characteristics** list, select the information that you want to view about this instance. The options are:
   - **LDAP Repository Database Sessions**—Selecting this option displays two graphs—one for open database sessions, the other for active database sessions at the end of the specified time period of statistics collection.
   - **Response Time vs. LDAP Operations**—Selecting this option displays two graphs. The first shows the average LDAP operation response time over the course of the specified time period of statistics collection. The other shows the number of operations in progress at the end of that period
   - **Active LDAP Sessions vs. New LDAP Sessions**—Selecting this option displays two graphs. The first shows the number of active LDAP sessions—that is, those that remain open at the end of the specified time period of statistics collection. The second shows new LDAP sessions—that

is, those that are opened over the course of the specified time period of statistics collection.

4. When you have made your selection, choose **Go**.

Viewing Directory Server Operations by Using Oracle Enterprise Manager Application Server Control

You can view directory server operations over the course of the specified time period of statistics collection by using Application Server Control. To do this:

1. In the Directory Server main window, select the directory server instance whose information you want to view.
2. Choose **View Operations**. This displays charts for all of the LDAP operations. Click any chart to see a larger image of it.

11

Backup and Restoration of a Directory

This chapter tells how to backup and restore both small and large directories. It contains these topics:

Backing Up and Restoring a Small Directory or Specific Naming Context

To backup and restore a small directory or specific naming context in directory, do the following:

1. Backup the node by using the ldifwrite utility. Enter this command:

   ```
   ldifwrite -connect connect_string  -b naming_context -f backup.ldif
   ```
2. Start the directory server on the new node by entering this command:

   ```
   oidctl connect= connect_string  server=oidldapd instance=1
   flags= '-p port_number' start
   ```
3. Load data into the new node by using the ldapaddmt utility. Enter this command:

   ```
   ldapaddmt -h host_name -p port_number -v -f backup.ldif
   ```

Backing Up and Restoring a Large Directory

To backup and restore a very large directory, do the following:

1. Stop the directory server on the backup node by entering this command:

   ```
   oidctl connect= connect_string  server=oidldapd instance=1 stop
   ```
2. Backup the sponsor node by using the export utility. To do this:
   a. Create a new file, `params1.dat`, containing the following:

      ```
      FILE=odsschema.imp
      OWNER=ods, odscommon
      GRANTS=y
      ROWS=y
      ```
   b. Run the following command against the identified sponsor node:

      ```
      exp system/manager FILE=output_file_name PARFILE=params1.dat
      ```

      Note: The directory schema and data are backed up in the `odsschema.imp` file. Move this file to the new node before performing next task.
3. Load Data into the new node by using the import utility. To do this:
   a. Be sure that the backup `odsschema.imp` file is present in current directory.
   b. Run the following SQL scripts:

      ```
      cd $ORACLE_HOME/ldap/admin/
      ```

```
sqlplus system/manager @ldapxact.sql
sqlplus system/manager @ldapxsec.sql
```

c. Create a new file, `params2.dat`, containing the following:

```
FILE=odsschema.imp
FROMUSER=ods
TOUSER=ods
```

d. Run the following commands against the new node:

```
imp system/manager FILE=input_file_name PARFILE=params2.dat
```

e. Create a new file, params3.dat, containing the following:

```
FILE=odsschema.imp
FROMUSER=odscommon
TOUSER=odscommon
```

f. Run the following command against the new node:

```
imp system/manager FILE=input_file_name PARFILE=params3.dat
```

4. Start the directory server on the new node by entering this command:

```
oidctl connect= connect_string server=oidldapd instance=1
flags='-p port_number' start
```

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:

12

Directory Security Concepts

Oracle Internet Directory is a key element of the Oracle Identity Management Infrastructure. This enables you to deploy multiple Oracle components to work against a shared instance of Oracle Internet Directory and associated infrastructure pieces. This sharing allows an enterprise to simplify security management across all applications.

In addition to the role it plays in the Oracle Identity Management infrastructure, Oracle Internet Directory provides many powerful features for protecting information.

This chapter gives a conceptual overview of Oracle Internet Directory security features. It contains these topics:

Data Integrity and Oracle Internet Directory

Oracle Internet Directory ensures that data has not been modified, deleted, or replayed during transmission by using Secure Sockets Layer (SSL). SSL 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 13, "Secure Sockets Layer (SSL) and the Directory" for more information about SSL

Data Privacy and Oracle Internet Directory

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 40 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 13, "Secure Sockets Layer (SSL) and the Directory" for more information about SSL

Authorization in Oracle Internet Directory

Authorization is the permission given to a user, program, or process to access an object or set of objects. When directory operations are attempted within a directory session, the directory server ensures that the user has the permissions to perform

those operations. If the user does not have the permissions, then the directory server disallows the operation. The directory server protects directory data from unauthorized operations by directory users by using access control information.

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 represent the permissions that various directory user entities (or subjects) have on a given object.

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 hosted companies. The realm 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 realm can allow only a user's manager to update that user's salary attribute. The administrator of another realm can establish and enforce a different policy regarding salary attributes.

Authentication in Oracle Internet Directory

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 three general kinds of authentication, and these are described in these topics:

- Direct Authentication
- Indirect Authentication
- External Authentication

Direct Authentication

This section describes the three kinds of authentication available within Oracle Internet Directory, and about how SASL-enabled clients authenticate to a directory server.

Direct Authentication Options

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.

- Authentication by using Simple Authentication and Security Layer (SASL)

  This is a method for adding authentication support to connection-based protocols. To use SASL, a protocol includes a command for identifying and authenticating a user to a server and for optionally negotiating protection of subsequent protocol interactions. If the use of SASL is successfully negotiated, then a security layer is inserted between the protocol and the connection.

  Oracle Internet Directory supports two authentication mechanisms with SASL:

  – MD5Digest—Within LDAP Version 3, this is the required authentication mechanism (RFC 2829). It uses the MD5 hash function to convert a message of any length to a 128 bit message digest that can be used as a verifier for client/server authentication.

  – External authentication—for example, SSL. In this case, the client, in lieu of a user name and password, authenticates to the server by means of a certificate, token or some other device as required by the external authentication mechanism.

  See Also:

  - Authentication by Using Simple Authentication and Security Layer (SASL) on page 12-9
  - The Web site of the Internet Engineering Task Force (IETF) at http://www.ietf.org for the following RFCs: RFC 2829, which specifies SASL Digest-MD5 as the required authentication mechanism for LDAP Version 3 servers; RFC 2831, which describes the Digest-MD5 mechanism; RFC 2617, which describes the HTTP Digest authentication mechanism on which SASL Digest-MD5 is based

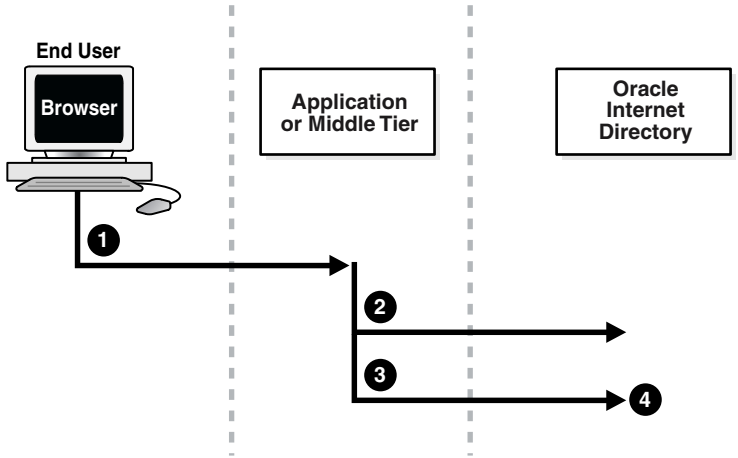
Indirect Authentication

Indirect authentication occurs through any entity that has credentials in the directory—for example, an application such as the Oracle Internet Directory Self-Service Console, or a middle tier such as a firewall or a RADIUS server. The application or middle tier becomes a **proxy user**. A proxy user has the privilege to

impersonate an end user, performing on that user's behalf those operations for which that user has privileges.

Figure 12–1 and the accompanying text explain how indirect authentication takes place.

Figure 12–1 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 end user entry does give the application or middle tier 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 end user entry does not give the application or middle tier the necessary proxy right, then the directory server returns an "Insufficient Access" error message.

See Also: Operations: What Access Are You Granting? on page 14-11

The directory server can, in the same session, authenticate and authorize other end users. It can also switch the session from the end user to 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 the 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 directory server verifies that `cn=user1` has proxy access, it allows this second bind to succeed. It does not require any further validation of the end-user DN, such as a password. For the rest of the session, all LDAP operations are access-controlled as if `cn=User2` were performing them.

If one user is being serviced by an application, and another user subsequently requests a service of that same application, then the application can establish a new connection and proceed as previously described without disrupting that prior session. If, however, no prior user is still being serviced, then the existing established connection can be re-used again and again without the need for a new connection.

External Authentication

Perhaps your enterprise stores user security credentials in a repository other than Oracle Internet Directory—for example, a database or another LDAP directory. With Oracle Internet Directory external authentication and password modification plug-ins, you can use these credentials for user authentication to Oracle components. You do not need to store the credentials in Oracle Internet Directory and then worry about keeping them synchronized.

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 and Managing Password Verifiers for Authenticating to Oracle Internet Directory"

Password Policies in Oracle Internet Directory

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 15, "Password Policies in Oracle Internet Directory" for a fuller description of the rules you set when establishing password policies

Authentication by Using Simple Authentication and Security Layer (SASL)

The section "Direct Authentication" on page 12-4 introduced the use of SASL within an Oracle Internet Directory environment. This section describes more fully how SASL works. It contains these topics:

- How a SASL-Enabled Client Authenticates to a Directory Server by Using Digest-MD5
- How a SASL-Enabled Client Authenticates to a Directory Server by Using External Authentication

How a SASL-Enabled Client Authenticates to a Directory Server by Using Digest-MD5

When a SASL-enabled client seeks Digest-MD5 authentication to a server, the authentication process is as follows:

1. The directory server sends to the LDAP client data that includes various authentication options that it supports and a special token.
2. The client selects an authentication option, then sends response to the server indicating the option it has selected. The response is encrypted so as to prove that the client knows its password.
3. The directory server then decrypts and verifies the client response.

How a SASL-Enabled Client Authenticates to a Directory Server by Using External Authentication

Oracle Internet Directory provides SASL-external authentication over an SSL connection in which both client and server authenticate themselves to each other by providing certificates. The DN is derived from the client certificate used in the SSL network negotiation.

When a client seeks authentication to a directory server by using an external authentication mechanism such as SSL, the authentication process is as follows:

1. The client sends an initial message with the authorization identity.
2. The directory server uses information external to SASL to determine whether the client can validly authenticate as the authorization identity. If the client can validly authenticate, then the directory server indicates successful completion of the authentication exchange. Otherwise, the directory server indicates failure.

The system providing the external information may be IPsec or SSL/TLS. If the client sends an empty string as the authorization identity, then the authorization identity is derived from the client authentication credentials in the system providing external authentication—for example, the SSL certificate.

13

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
- Limitations of the Use of SSL in Oracle Internet Directory 10g (9.0.4)

> **See Also:** "Security" on page 2-11 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 13–1 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 389.

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 Schema Elements" on page B-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.

Adding a New SSL Configuration Set

Note: Prior using Oracle Directory Manager to add a new SSL configuration set, you must do the following by using Oracle Wallet Manager:

- Create a new wallet
- Create a certificate request and send it to your certificate authority
- If your certificate authority is not included in the default list of trusted certificates in Oracle Wallet Manager, then import the trusted certificate of your certificate authority into your wallet
- Save the wallet with auto-login enabled

See Also: The chapter on Oracle Wallet Manager in *Oracle Advanced Security Administrator's Guide*

To add a new SSL configuration set:

1. In the navigator pane, expand in succession Oracle Internet Directory **Servers,** ***directory server instance***, **Server Management**.
2. Expand either **Directory Server** or **Replication Server**, as appropriate. The numbered configuration sets are listed beneath your selection.
3. Select the default configuration set.
4. Choose **Create Like**. The Configuration Sets dialog box displays the **General** tab page.
5. In the **General** tab page, change the value of the non-SSL port to something other than the default (389 or 4032).
6. Select the **SSL Settings** tab, and, enter values in the appropriate fields. These fields are described in Table C–33 on page C-28.

Viewing and Modifying SSL Configuration Parameters

To view and modify SSL configuration parameters:

1. In the navigator pane, expand in succession Oracle Internet Directory **Servers,** ***directory server instance***, **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, modify the fields and save the changes. These fields are described in Table C–39 on page C-37.

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 5-7
- "Entry and Attribute Management Command-Line Tools Syntax" on page A-18 for instructions on using the `-p`, `-U`, and `-W` flags to configure SSL

Starting a Directory Server Instance with SSL Enabled

Note: Beginning with Oracle Internet Directory Release 9.0.2, only wallets in encrypted (`cwallet.sso`) format are supported. This means that, before you can start an SSL instance, you must use Oracle Wallet Manager enabled to AutoLogin to open the wallet. In releases of Oracle Internet Directory prior to Release 9.0.2, a password can be provided to open the wallet.

On the Windows operating system, before starting a directory server instance with SSL enabled, you must change the Logon Account of the Oracle Directory Service from Local System Account to the user who owns the wallet. This user should be member of the Administrator Group.

To change the services:

- On Windows 2000: Choose in succession **Start**, **Settings**, **Control Panel**, **Administrative Tools**, **Services**.
- On Windows NT: Choose in succession **Start**, **Settings**, **Control Panel**, **Services**.

Right-click **Oracle Directory Service**, then choose **Properties**.

Select the **Logon** tab.

Clear the **Local System Account** radio button, and select **This Account**.

Enter the account you logged in as when you created the wallet.

Stop and restart the service.

Use the OID Control utility to start the directory server with SSL enabled.

In this example, SSL is configured in Configset 1. It is assumed that no directory server instance with instance ID 2 is running. Enter this command:

```
oidctl connect=<et_service_name server=oidldapd instance=2 configset=1 start
```

Limitations of the Use of SSL in Oracle Internet Directory 10*g* (9.0.4)

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 10*g* (9.0.4), 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

14

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
- How ACL Evaluation Works
- Managing Access Control by Using Oracle Directory Manager
- Managing Access Control by Using Command-Line Tools

See Also:

- "Security" on page 2-11 and Chapter 12, "Directory Security Concepts" for a conceptual explanation before you begin implementing and administering access control policies
- Appendix E, "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 Level Requirements for LDAP Operations

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 14-13

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". 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.

> **See Also:** "Object: To What Are You Granting Access?" on page 14-7 for the structure definition of the orclEntryLevelACI attribute

Security 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 security groups. There are two types of security 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 realm. Because the global administrator's identity is not recognized in the realm of the hosted company, the directory server, relying only on the ACPs in that realm, 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 to this privilege

group the access rights to that subtree. If it finds such an ACP, then the directory server overrides the denials by ACPs in the hosted company's realm.

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 do the following:

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 Security 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 security 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 security 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 security groups. This process continues until there are no more nested groups to be evaluated.

Each security group, nested or otherwise, must be associated with an security group object class—either `orclACPgroup` or `orclPrivilegeGroup`. Even if a group is a member of an security group, the directory server does not consider it for access control purposes unless it is associated with an security group object class. When it has determined the user's membership in security groups, the directory server uses that information for the lifetime of the session.

Example: Computing Security 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 3,`c=us` contains the following nested groups:

- `cn=group2,c=us`
- `cn=group1,c=us`
- `cn=group4,c=us`

Access control policies for Group 3 are applicable to members of Group 3, Group 1, and Group 2 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 Group 3 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 Group 2—then his access rights will be governed by access policies set up for members of Group 1, Group 2, as well as Group 3 (in which Group 1 and Group 2 are nested). This is because all three groups are associated with the object class `orclPrivilegeGroup`.

> **See Also:** Either "Modifying Entries by Using Oracle Directory Manager" on page 7-7 or "Example: Modifying a User Entry by Using ldapmodify" on page 7-12 for instructions on how to modify a group entry to associate it with or disassociate it from either the `orclPrivilegeGroup` or the `orclACPgroup` object class

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-delimited list of attribute names in the object selector.

```
attr=(attribute_list)
```

Attributes within an entry are excluded from a policy by including a comma-delimited 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 E, "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 methods of authentication and of encryption to be used by the subject.

There are four authentication modes:

- MD5Digest
- PKCS12
- Proxy
- Simple: Simple password-based authentication

There are three encryption options:

- SASL
- SSL No Authentication
- SSL One Way

Specifying the encryption mode is optional. If it is not specified, then no encryption is used—unless the selected authentication mode is PKCS12. Data transmitted by using PKCS12 is always encrypted.

There is a precedence rule among authentication choices, and it is as follows:

Anonymous < Proxy < Simple < MD5Digest < PKCS12

This rule means that:

- Proxy authentication blocks anonymous access
- Simple authentication blocks both Proxy and Anonymous access
- MD5Digest authentication blocks Simple, Proxy and Anonymous access
- PKCS12 authentication blocks MD5Digest, Simple, Proxy and Anonymous access

The bind mode syntax is:

```
BINDMODE =(LDAP_AUTHENTICATION_CHOICE + [ LDAP_ENCRYPTION_CHOICE ] )
LDAP_AUTHENTICATION_CHOICE = Proxy | Simple | MD5Digest | PKCS12
LDAP_ENCRYPTION_CHOICE = SSLNoAuth | SSLOneway | SASL
```

The *LDAP_ENCRYPTION_CHOICE* is an optional parameter. If you do not specify it, then the directory server assumes that no encryption is to be used.

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 Also:** Appendix E, "The Access Control Directive Format" and Appendix D, "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.

Table 14–1 Types of Access

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

Table 14–1 (Cont.) Types of Access

Access Level	Description	Type of Object
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.

Access Level Requirements for LDAP Operations

The following table lists LDAP operations and the access required to perform each one.

Table 14–2 LDAP Operations and Access Needed 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 and Browse access to the entry
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)

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:

Table 14–3 Attribute States During ACL Evaluation

State	Description
Resolved with permission	The required access for the attribute has been granted in the ACI.

Table 14–3 (Cont.) Attribute States During ACL Evaluation

State	Description
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.

This section contains these topics:

- Precedence Rules Used in ACL Evaluation
- Use of More Than One ACI for the Same Object
- Exclusionary Access to Directory Objects
- ACL Evaluation For Groups

Precedence Rules Used in ACL Evaluation

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)
   ```

Use of More Than One ACI for the Same Object

Beginning with Release 9.0.4, Oracle Internet Directory, enables you to define more than one ACI in the ACP of an object. Oracle Internet Directory processes the ACIs associated with that object and stores them as a single ACI in its internal ACP cache. It then applies all the relevant policies in the multiple ACIs specified in the ACP.

The following example of an ACP illustrates how tis works.

```
Access to entry by dn="cn=john" (browse,noadd,nodelete)
Access to entry by group="cn=admingroup" (browse,add,nodelete)
Access to entry by dn=".*,c=us" (browse,noadd,nodelete)
```

In this ACP, there are three ACIs for the object entry. When it loads this ACP, Oracle Internet Directory merges these three ACIs as one ACI in its internal ACP cache.

The ACI syntax is:

```
Access to <OBJECT> by <SUBJECT> <ACCESSLIST>
<OBJECT> = [ entry |  attr [EQ-OR-NEQ] ( * | <ATTRLIST> ) ]
[ filter = ( <LDAFILTER> ) ]
```

This syntax makes possible the following types of objects:

- Entry
- Entry + filter = (*LDAPFILTER*)
- Attr = (*ATTRLIST*)
- Attr = (*ATTRLIST*) + filter = (*LDAPFILTER*)
- Attr != (*ATTRLIST*)
- Attr != (*ATTRLIST*) + filter = (*LDAPFILTER*)

- Attr = (*)
- Attr = (*) + filter = (*LDAPFILTER*)

You can define multiple ACIs for any of the above types of objects. During initial loading of the ACP, the directory server merges the ACIs based on which of these object types are defined. The matching criterion is the exact string comparison of the object strings in the ACIs.

If one ACI specifies `ATTR=(ATTRLIST)` and another `ATTR!=(ATTRLIST)`, then `ATTR=(*)` must not be specified as an ACI in the entry. Also, if an ACI specifies ATTR=(ATTRLIST), then, to specify the access rights to attributes not in ATTRLIST, ATTR=(*) must be used and not ATTR!=(ATTRLIST). ATTR=(*) implies all attributes other than those specified in ATTRLIST.

Exclusionary Access to Directory 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 that object 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 DIT.

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.

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 3-2

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. Choose **OK**.
6. To effect your changes, 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. Choose **OK**. A notice window displays the message "You need to restart Oracle Directory Manager to view ACP Management Changes."
7. Choose OK for the Notice window.
8. To view the latest Access Control Management entries, disconnect and immediately reconnect Oracle Directory Manager.

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 14-18, then you can locate and view an ACP as follows:

1. In the navigator pane, expand in succession **Oracle Internet Directory Servers**, ***directory server instance***, **Access Control Management**. All of the defined ACPs appear in the navigator pane below the Access Control Management node.
2. In the navigator pane, under **Access Control Management**, select an ACP to display its information in the right pane. The fields in the Access Control Management pane are described in Table C–1 on page C-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 14-18, then you can locate and view an ACP as follows:

1. Expand in succession **Oracle Internet Directory Servers**, ***directory server instance***, then select **Entry Management**.
2. 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.
3. In the **Distinguished Name** box, double-click the entry. The corresponding Entry dialog box appears.
4. 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 14-18, then begin as follows:

   a. In the navigator pane, expand in succession **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 14-18, then begin as follows:

   a. In the navigator pane, expand in succession **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, choose **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 choosing **Browse** to the right of the **Path to Entry** field.

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, choose **Create**. The Structural Access Item dialog box appears. It has four tabs: **Entry Filter**, **Added Object Filter**, **By Whom**, and **Access Rights**.

2. In an ACP, the access rights defined apply to the entry and all its subentries unless other filters restrict access further.

   If you want all entries below the ACP to be governed by the ACP, then you do not need to enter anything on the **Entry Filter** tab page; simply proceed to the next step. Otherwise, perform this step.

   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 search criteria bar, select an attribute type.

b. From the menu in the middle of the bar, select one of the filter options. These options are described in Table C–37 on page C-35.

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 search criteria bar, select an attribute type.

   b. From the menu in the middle of the bar, select one of the filter options. These options are described in Table C–37 on page C-35.

   c. In the text box at the right of the search criteria bar, type the value for the attribute you selected.

4. Select the **By Whom** tab page.

   a. From the **Authentication Choice** list, select the type of authentication to be used by the subject (that is, the entity that seeks access). The options are described in Table C–2 on page C-2.

      If you do not choose an authentication method, then any kind of authentication is accepted. The authentication method specified on one node should match the one specified on the node it is communicating with.

      From the **Encryption Choice** list, select the type of encryption to be used. The options are described in Table C–3 on page C-3.

   b. Specify the entity or entities to whom you are granting access. The options are described in Table C–4 on page C-3.

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, choose **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 Step 3. Otherwise, perform this 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 search criteria bar, select an attribute type.

   b. From the menu in the middle of the bar, select one of the filter options. These are described in Table C–37 on page C-35.

   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 **Authentication Choice** list, select the type of authentication to be used by the subject (that is, the entity that seeks access). The options are described in Table C–2 on page C-2.

   If you do not choose an authentication method, then any kind of authentication is accepted. The authentication method specified on one node should match the one specified on the node it is communicating with.

From the **Encryption Choice** list, select the type of encryption to be used. The options are described in Table C–3 on page C-3.

b. Specify the entity or entities to whom you are granting access. The options are described in Table C–4 on page C-3.

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 privileges. These are described in Table C–5 on page C-4.

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 14-18, then begin as follows:

   a. In the navigator pane, expand in succession **Oracle Internet Directory Servers** and **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 14-18, then begin as follows:

a. In the navigator pane, expand in succession **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.

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.

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.

2. 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.

   Alternatively, if you want to grant access to a specific entry, then, in this first Structural Access Item dialog box, do the following:

   a. From the menu at the left of the search criteria bar, select an attribute type.

   b. From the menu in the middle of the bar, select one of the filter options. These are described in Table C–37 on page C-35.

   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.

3. 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 search criteria bar, select an attribute type.

b. From the menu in the middle of the bar, select one of the filter options. These are described in Table C–37 on page C-35.

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 choose the authentication and encryption methods, and the subject to whom you are granting access.

4. Specifying the authentication method is optional. If you do not set an authentication method, then any kind of authentication is accepted. The authentication method specified on one node must match the bind mode specified on the node it is communicating with.

   a. To specify the type of authentication, from the **Authentication Choice** list, select the type of authentication to be used by the subject (that is, the entity that seeks access). The options are described in Table C–2 on page C-2.

   b. To specify the type of encryption, from the **Encryption Choice** list, choose an encryption method. The options are described in Table C–3 on page C-3.

   c. Specify the entity or entities to whom you are granting access. Options are described in Table C–4 on page C-3.

   d. Click **Next**. A Structural Access Item dialog box prompts you for access rights information.

5. 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

6. Click **Finish**.

Task 3: Configure Content Access Items by Using the ACP Creation Wizard

1. 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.
2. 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.

   Alternatively, to identify an attribute to which you are specifying access:

   a. From the menu at the left end of the search criteria bar, select an attribute type.
   b. From the menu in the middle of the bar, select one of the filter options. These are described in Table C–25 on page C-21.
   c. In the text box at the right end of the search criteria bar, type the value for the attribute you selected.
   d. Click **Next**. A second Content Access Item dialog box prompts you to specify to whom you are granting access.
   e. Choose **Next**. The wizard prompts you to choose the authentication and encryption methods, and the subject to whom you are granting access.
3. Specifying the authentication method is optional. If you do not set an authentication method, then any kind of authentication is accepted. The authentication method specified on one node must match the bind mode specified on the node it is communicating with.

   a. To specify the type of authentication, from the **Authentication Choice** list, select the type of authentication to be used by the subject (that is, the entity that seeks access). The options are described in Table C–2 on page C-2.
   b. To specify the type of encryption, from the **Encryption Choice** list, choose an encryption method. The options are described in Table C–3 on page C-3.
   c. Specify the entity or entities to whom you are granting access. Options are described in Table C–4 on page C-3.
   d. Click **Next**. A Content Access Item dialog box prompts you to select an attribute and the matching operation to be performed against it.
4. 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.

5. Specify what kinds of rights are granted. These are described in Table C–5 on page C-4.

6. 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 14-18, then begin as follows:

   a. In the navigator pane, expand **Oracle Internet Directory Servers**, ***directory server instance***, and **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 14-18, then begin as follows:

   a. In the navigator pane, expand in succession **Oracle Internet Directory Servers**, ***directory server instance***, and **Access Control Management**.

   b. Select the ACP you want to modify. The information for that ACP is displayed in the right pane.

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 14-21 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 search criteria bar, select an attribute.
   b. From the menu in the middle of the bar, select one of the filter options. These are described in Table C–25 on page C-21.
   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 restricting 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 search criteria bar, select an attribute type.

b. From the menu in the middle of the bar, select one of the filter options. These are described in Table C–37 on page C-35.

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 authentication and encryption methods, and the subject of the ACI (that is, the entity that seeks access).

   Specifying the authentication method is optional. If you do not set an authentication method, then any kind of authentication is accepted. The authentication method specified on one node must match the bind mode specified on the node it is communicating with.

   a. To specify the type of authentication, from the **Authentication Choice** list, select the type of authentication to be used by the subject (that is, the entity that seeks access). The options are described in Table C–2 on page C-2.

   b. To specify the type of encryption, from the **Encryption Choice** list, choose an encryption method. The options are described in Table C–3 on page C-3.

   c. Specify the entity or entities to whom you are granting access. The options are described in Table C–4 on page C-3.

5. Select the **Access Rights** tab page.

   a. Determine 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

     If an entry is unspecified, then access is determined at the next highest level in which access is specified.

6. 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 14-23 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 Items** box, 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 search criteria bar, select an attribute type.
   b. From the menu in the middle of the bar, select one of the filter options. These are described in Table C–37 on page C-35.
   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 **By Whom** tab page to specify the authentication and encryption methods, and the subject of the ACI (that is, the entity that seeks access).

   Specifying the authentication method is optional. If you do not set an authentication method, then any kind of authentication is accepted. The authentication method specified on one node must match the bind mode specified on the node it is communicating with.

   a. To specify the type of authentication, from the **Authentication Choice** list, select the type of authentication to be used by the subject (that is, the entity that seeks access). The options are described in Table C–2 on page C-2.
   b. To specify the type of encryption, from the **Encryption Choice** list, choose an encryption method. The options are described in Table C–3 on page C-3.
   c. Specify the entity or entities to whom you are granting access. The options are described in Table C–4 on page C-3.
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 privileges. These are described in Table C–5 on page C-4.

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 in succession **Oracle Internet Directory Servers**, ***directory server instance***, **Entry Management**. You may either:

   - In the navigator pane, select the entry to display its properties in the right pane
   - In the right pane, search for 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 as described in "Modifying an ACP by Using Oracle Directory Manager" on page 14-28.

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. Otherwise, the information 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 in succession **Oracle Internet Directory Servers**, ***directory server instance***.
2. Select **Access Control Management**. A list of ACPs appears in the right pane.
3. At the bottom of the right pane, click **Create**. A New Access Control Point dialog box appears.
4. 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 these tabs: **Entry Filter**, **Added Object 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 **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:

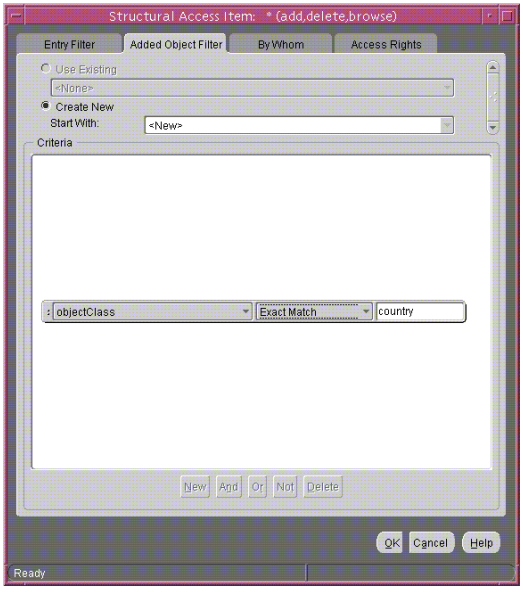
   a. From the menu at the left end of the search criteria bar, select the `objectclass` attribute type.

   b. From the menu in the middle of the bar, select **Exact Match**.

   c. In the text box at the right of the search criteria bar, enter `country`.

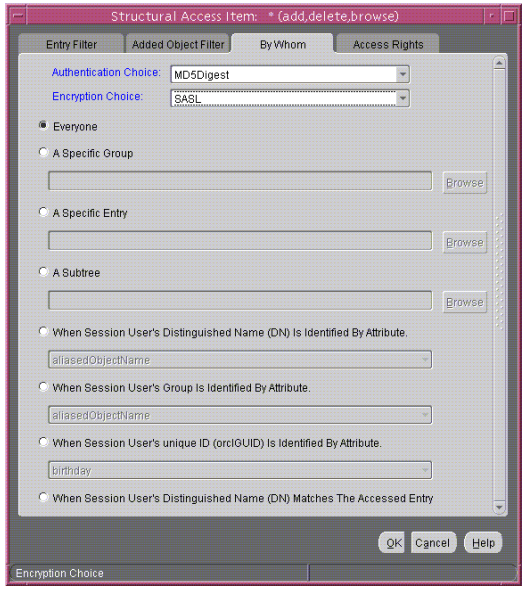
The **Added Object Filter** tab page should now look like Figure 14–1.

Figure 14–1 Structural Access Item: Added Object Filter Tab Page



3. Select the **By Whom** tab page.
   a. From the **Authentication Choice** list, select **MD5Digest**.
   b. From the **Encryption Choice** list, choose **SASL**.
   c. To create access rights for everyone, select **Everyone**. The **By Whom** tab page should look like Figure 14–2.

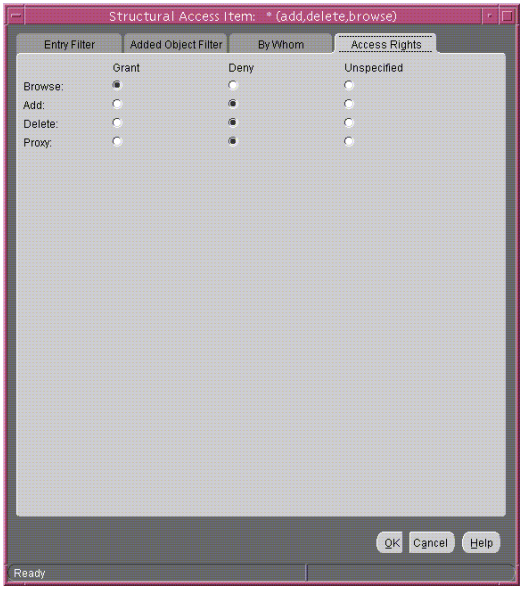
Figure 14–2 Structural Access Item: By Whom Tab Page



4. 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. The **Access Rights** tab page should look like Figure 14–3.

Figure 14–3 Example: Structural Access Item: Access Rights Tab Page



b. Click **OK**.

Configure Content Access Items The four ACIs in this example use the same structural access 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.

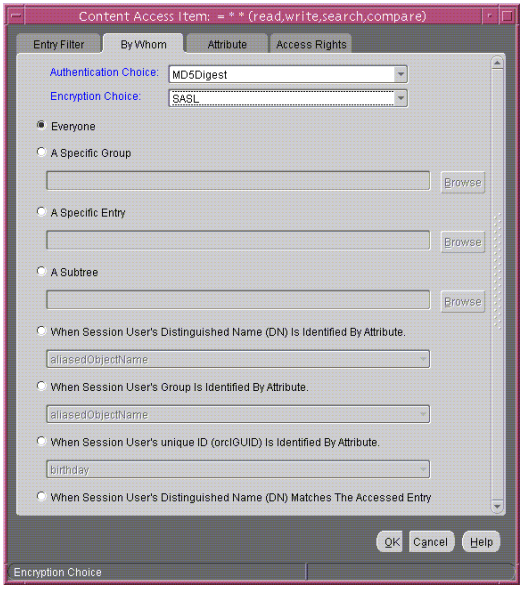
2. Select the **By Whom** tab page.

   a. From the **Authentication Choice** list, select **MD5Digest**.

b. From the **Encryption Choice** list, choose **SASL**.

c. To create access rights for everyone, select **Everyone**. The **By Whom** tab page should look like Figure 14–4.

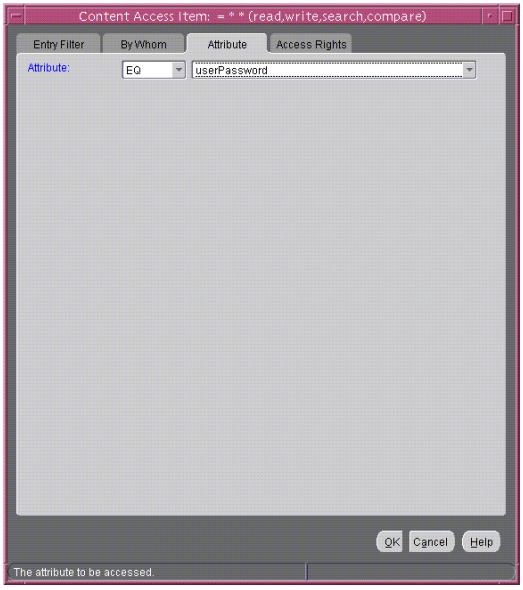
Figure 14–4 Content Access Item: By Whom Tab Page



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`**. The **Attribute** tab page should look like Content Access Item: Attribute Tab PageFigure 14–5.

Figure 14–5 Content Access Item: Attribute Tab Page



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. The **Access Rights** tab page should look like Figure 14–6.

Figure 14–6 Content Access Item: Access Rights Tab Page

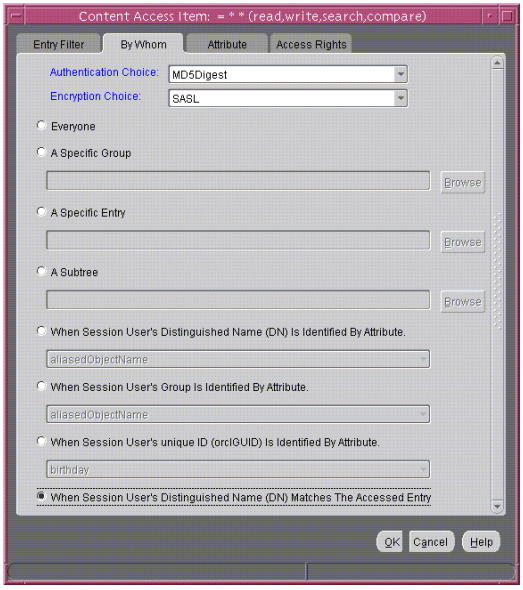
5. Click **OK**.

   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.
   a. From the **Authentication Choice** list, select **MD5Digest**.
   b. From the **Encryption Choice** list, choose **SASL**.
   c. To create access rights for everyone, select **When Session User's Distinguished Name (DN) Matches the Accessed Entry**. The By Whom tab page should look like Figure 14–7.

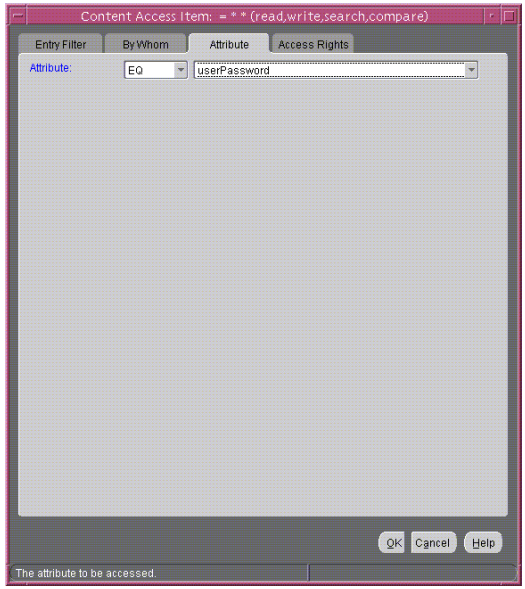
Figure 14–7 Content Access Item: By Whom Tab Page



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`**. The **Attribute** tab page should look like Figure 14–8.

Figure 14–8 Content Access Item: Attribute Tab Page



4. Select the **Access Rights** tab page.

   Grant access to read, search, write, and compare. Leave selfwrite unspecified. The **Access Rights** tab page should look like Figure 14–9.

Figure 14–9 Content Access Item: Access Rights Tab Page

5. Click **OK**.

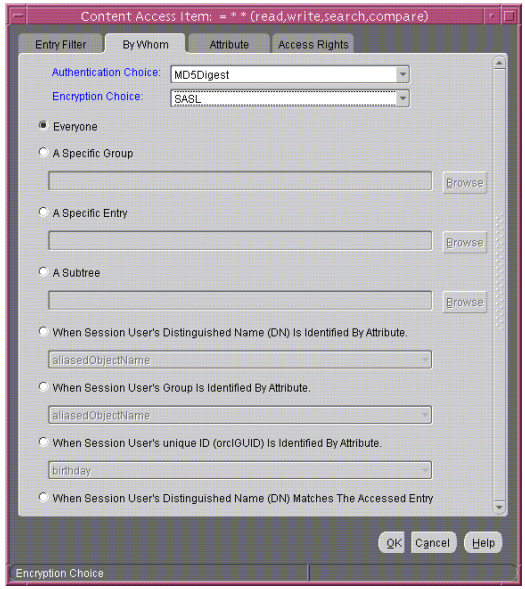
You have now created two ACIs. One denies Everyone read, search, write, and compare access to the `userPassword` attribute. The second allows the owner of the password to read, search, write, and compare that attribute.

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** box, click **Create** to display the Content Access Items dialog box.
2. Select the **By Whom** tab page.
   a. From the **Authentication Choice** list, select **MD5Digest**.
   b. From the **Encryption Choice** list, choose **SASL**.
   c. To create access rights for everyone, select **Everyone**. The **By Whom** tab page should look like Figure 14–10.

Figure 14–10 Content Access Item: By Whom Tab Page

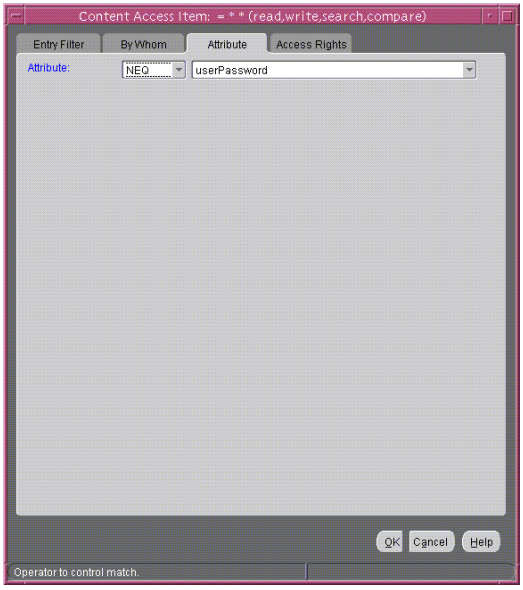


3. Select the **Attribute** tab page.

   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. The **Attribute** tab page should look like Figure 14–11.

Figure 14–11 Content Access Item: Attribute Tab Page



4. Select the **Access Rights** tab page.

   Grant access to read, search, and compare. Deny write access. Leave selfwrite unspecified. The **Access Rights** tab page should look like Figure 14–12.

Figure 14–12 Content Access Item: Access Rights Tab Page

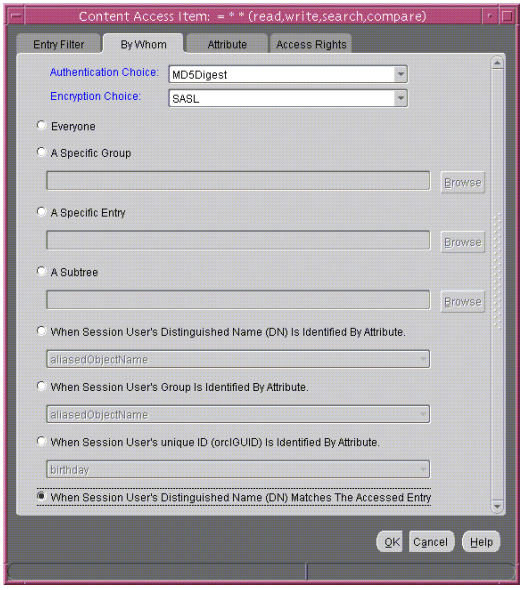
5. Click **OK** to apply these permissions and close the dialog box.

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 Item**s box, click **Create** to display the Content Access Items dialog box.
2. Select the **By Whom** tab page.
   a. From the **Authentication Choice** list, select **MD5Digest**.
   b. From the **Encryption Choice** list, choose **SASL**.
   c. To create access rights for everyone, select **When Session User's Distinguished Name (DN) Matches the Accessed Entry**. The By Whom tab page should look like Figure 14–13.

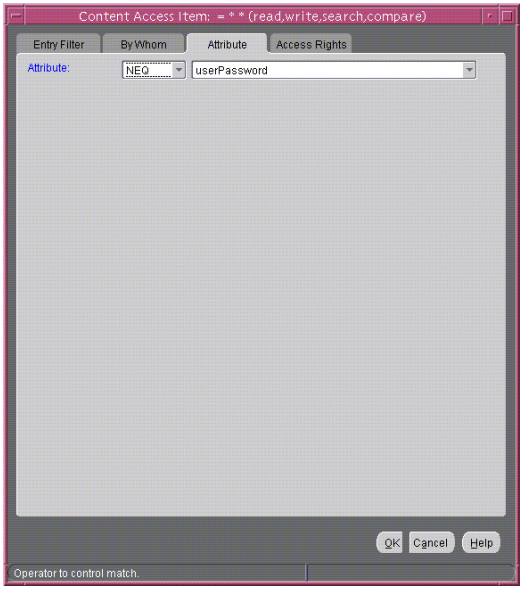
Figure 14–13 Content Access Item: By Whom Tab Page



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. The **Attribute** tab page should look like Figure 14–14.

Figure 14–14 Content Access Item: Attribute Tab Page



4. Select the **Access Rights** tab page.

   Grant access to read, search, and write. Leave selfwrite unspecified. The **Access Rights** tab page should look like Figure 14–15

Figure 14–15 Access Rights Tab Page

5. Click **OK** to apply these permissions and close the dialog box.

Managing Access Control by Using Command-Line Tools

As described in "Overview of Access Control Policy Administration" on page 14-2, 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 E, "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-31 for information about how to run ldapmodify
- Appendix E, "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)
```

15

Password Policies in Oracle Internet Directory

This chapter discusses password policies—that is, sets of rules that govern how passwords are used.

This chapter contains these topics:

About Password Policies

This section contains these topics:

What a Password Policy Is

Password polices are sets of rules that govern how passwords are used. They can specify, for example:

- 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
- That users change their passwords periodically
- That users cannot reuse previously used passwords
- That users are locked out after a certain number of login attempts

Default Password Policy

The default password policy for Oracle Internet Directory enforces:

- Password expiration in 60 days
- Account lockout after 10 login failures. Except for the super user account, all accounts remain locked for a duration of 24 hours unless the passwords are reset by the directory administrator.

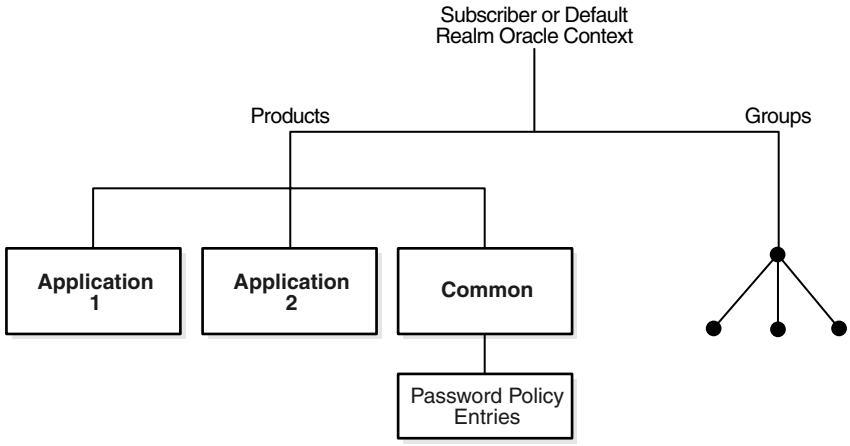
  If a super user account becomes locked, it stays locked until it is unlocked by using the OID Database Password utility. This utility prompts you for the ODS user password. After you enter the ODS password, it unlocks the account.
- A minimum password length of five characters with at least one numeric character
- A maximum of three grace logins after password expiration

Beginning in Oracle Internet Directory, Release 9.0.4, the password policy entry in the Root Oracle Context applies to the super user, but only the password policy governing account lockout is enforced on that account.

During Oracle Internet Directory installation, the Oracle Universal Installer creates for each identity management realm a password policy entry. This entry contains all password policy information applicable to all users in that realm.

The installer places this entry as shown in Figure 15–1—namely, immediately below the `common` entry, which resides under the `products` entry, which, in turn, resides under the Oracle Context specific to the identity management realm.

Figure 15–1 Location of Password Policy Entries



The Oracle Internet Directory password policy is applicable to simple binds (based on the `userpassword` attribute), compare operations on the `userpassword` attribute, and SASL binds. It does not apply to SSL and proxy binds.

To enforce this password policy, set to the appropriate value the `orclcommonusersearchbase` attribute in the `common` entry of the realm-specific Oracle Context. Otherwise, no password policy modification can take effect.

Directory Server Verification of Password Policy Information

To ensure that the user password meets the requirements of a given policy, the directory server verifies:

- That the password policy is enabled. It does this by checking the value of the attribute `orclpwdpolicyenable` in the password policy entry. A value of 1 indicates that the password policy is enabled. A value of 0 indicates that it is disabled.
- Correctness of password policy syntax information, which includes, for example, the correct number of alphabetic and numeric characters, or the correct password length. The directory server checks the syntax during `ldapadd` and `ldapmodify` operations.
- Password policy state information, which includes, for example:
  - The timestamp of the user password creation or modification
  - The timestamp of consecutive failed login attempts by the user
  - The time at which the user account was locked
  - Indicator that the password has been reset and must be changed by the user on first authentication
  - A history of user's previously used passwords
  - Time stamps of grace logins

  The directory server checks the state information during `ldapbind` and `ldapcompare` operations, but does so only if the `orclpwdpolicyenable` attribute is set to 1.

  To enable password value syntax checking, set the attributes `orclpwdpolicyenable` and `pwdchecksyntax` in the password policy entry to `TRUE`.

Overview: Establishing a Password Policy for an Identity Management Realm

In general, to establish a password policy:

1. Create a password policy entry, associate it with the `pwdpolicy` object class, and populate the corresponding attributes.
2. Set values for the `pwdPolicy` object class, which contains password policy information for the entire directory. Do this during installation when the entry of this object class is created.

3. Verify that the `orclpwdpolicyenable` attribute in the password policy entry is set to 1.

   See Also: "Password Policy Schema Elements" on page B-25 for a list and descriptions of the attributes of the `pwdPolicy` object class, and those of the `top` object class that pertain to password policies

Managing Password Policies

This section contains these topics:

- Managing Password Policies by Using Oracle Directory Manager
- Managing Password Policies by Using Command-Line Tools
- Managing Password Policies by Using the Self-Service Console

Table 15–1 lists the administrative tasks related to password policies, the tools you use to perform each one, and points you to the corresponding information.

Table 15–1 Tasks and Tools for Managing Password Polices

Task	Tools	Instructions
Enabling and disabling accounts	Oracle Internet Directory Self-Service Console ldapmodify	"Enabling and Disabling Accounts by Using the Oracle Internet Directory Self-Service Console" on page 15-10 "Example: Enabling and Disabling Accounts by Using Command-Line Tools" on page 15-9
Forcing a password change	ldapmodify	"Example: Forcing a Password Change by Using Command-Line Tools" on page 15-10
Modifying password policies for an identity management realm	Oracle Directory Manager ldapmodify	"Modifying Password Policies of an Identity Management Realm by Using Oracle Directory Manager" on page 15-7 "Example: Modifying Password Policies of an Identity Management Realm by Using Command-Line Tools" on page 15-9

Table 15–1 Tasks and Tools for Managing Password Polices

Task	Tools	Instructions
Setting password policies	ldapmodify	"Example: Setting Password Policies by Using Command-Line Tools" on page 15-8
Unlocking accounts	Oracle Internet Directory Self-Service Console ldapmodify	"Unlocking Accounts by Using the Oracle Internet Directory Self-Service Console" on page 15-10 "Example: Unlocking Accounts by Using Command-Line Tools" on page 15-9
Viewing password policies for an identity management realm	Oracle Directory Manager ldapsearch	"Viewing Password Policies of an Identity Management Realm by Using Oracle Directory Manager" on page 15-6 "Example: Viewing Password Policies of an Identity Management Realm by Using Command-Line Tools" on page 15-8

Managing Password Policies by Using Oracle Directory Manager

When you create the base entry for an identity management realm—whether during an Oracle Internet Directory installation or later—you also create a password policy entry for that realm. Later, you can use Oracle Directory Manager to view, refresh, and modify those policies.

This section contains these topics:

- Viewing Password Policies of an Identity Management Realm by Using Oracle Directory Manager
- Modifying Password Policies of an Identity Management Realm by Using Oracle Directory Manager

Viewing Password Policies of an Identity Management Realm by Using Oracle Directory Manager

To view the password policies for a particular identity management realm, in the navigator pane, expand in succession Oracle Internet Directory **Servers**, ***directory server instance***, **Password Policy Management**. The navigator pane displays the

password policy entries for the identity management realm. 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

To get the latest updates to realm-specific password policies, choose **Refresh**.

To get the password polices of a particular realm, in the navigator pane, choose the realm-specific password policy you want to view. The policies appear in the right pane.

> **See Also:** "Password Policy Fields in Oracle Directory Manager" on page C-6 for a description of each password policy displayed in Oracle Directory Manager

Modifying Password Policies of an Identity Management Realm by Using Oracle Directory Manager

To modify the password policies for a particular identity management realm:

1. In the navigator pane, expand in succession **Oracle Internet Directory Servers**, ***directory server instance***, **Password Policy Management**.
2. In the navigator pane, choose the realm-specific password policy you want to modify. The corresponding tab pages appear in the right pane.
3. In the **General** tab page, modify the editable attribute fields as needed. These fields are described in Table C–8 on page C-6.
4. Select the **Account Lockout** tab page and, to modify the fields, select **Global Lockout**. Modify the editable attribute fields as needed. These fields are described in Table C–9 on page C-8.
5. Select the **IP Lockout** tab page and, to modify the fields, select **IP Lockout**. Modify the editable attribute fields as needed. These fields are described in Table C–10 on page C-8.
6. Select the **Password Syntax** tab page and, to modify the fields, select **Check Password Syntax**. Modify the editable attribute fields as needed. These fields are described in Table C–11 on page C-8.
7. When you are finished, choose **Apply**.

Managing Password Policies by Using Command-Line Tools

This section contains these topics:

Example: Setting Password Policies by Using Command-Line Tools

The following example disables the `pwdLockout` attribute, changing it from its default setting of `1`.

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: 0
```

The following command loads this file into the directory:

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

Examples: Managing the Password Policies of an Identity Management Realm by Using Command-Line Tools

Look at the following examples to learn how to view and modify the password policies of a realm by using command-line tools.

Example: Viewing Password Policies of an Identity Management Realm by 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 Password Policies of an Identity Management Realm by 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
```

Example: Enabling and Disabling Accounts by Using Command-Line Tools

You can temporarily disable a user's account, then enable it once again, by using command-line tools.

To permanently disable the account by setting the `orclisenabled` attribute to `DISABLED`. Setting this attribute to any other value enables the account.

To enable the account after you have disabled it, delete this attribute from the entry.

To enable the account for a specific period, set the `orclActiveStartDate` and `orclActiveEndDate` attributes in the user entry to the proper value in **UTC (Coordinated Universal Time)** format. For example:

```
cn=John Doe,cn=users,o=my_company,dc=com
orclactivestartdate:20030101000000z
orclactiveenddate: 20031231000000z
```

In this example, John Doe can log in only between January 1, 2003 and December 31, 2003. He cannot login prior to January 1, 2003 or after December 31, 2003. If you want to disable his account for a period of time between these dates, then set the `orclisenabled` attribute to `FALSE`.

Example: Unlocking Accounts by Using Command-Line Tools

If you are a member of the Security Administrators Group, then, if an account becomes locked, you can unlock it without resetting the user password. This saves you from having to explicitly tell the user the new password. The user can simply log in by using the old password.

To unlock an account, set the `orclpwdaccountunlock` attribute to 1.

The following example unlocks the account for user John Doe.

```
ldapmodify -p port_number -h host_name -D cn=orcladmin -w welcome -v <<EOF
dn: cn=John Doe,cn=users,o=my_company,dc=com
changetype: modify
```

```
add: orclpwdaccountunlock
orclpwdaccountunlock: 1
```

Example: Forcing a Password Change by Using Command-Line Tools

You can force users to change their passwords when they log in for the first time. To do this, set the `pwdMustChange` attribute in the `pwdpolicy` entry to `TRUE`, and then reset the password. If you do this, you must explicitly tell the user the new password so that the user can login to change that password.

> **See Also:** "Resetting Your Own Password by Using the Oracle Internet Directory Self-Service Console" on page 15-11 for instructions on resetting passwords

Managing Password Policies by Using the Self-Service Console

This section contains these topics:

- Managing Accounts by Using the Oracle Internet Directory Self-Service Console
- Enabling and Disabling Accounts by Using the Oracle Internet Directory Self-Service Console
- Unlocking Accounts by Using the Oracle Internet Directory Self-Service Console
- Resetting Your Own Password by Using the Oracle Internet Directory Self-Service Console

Managing Accounts by Using the Oracle Internet Directory Self-Service Console

You can use the Oracle Internet Directory Self-Service Console to enable, disable, and unlock user accounts.

Enabling and Disabling Accounts by Using the Oracle Internet Directory Self-Service Console

You can temporarily disable a user's account, then enable it once again, by using the Oracle Internet Directory Self-Service Console.

> **See Also:** "Enabling User Accounts" on page 31-25 and "Disabling User Accounts" on page 31-25 for instructions on enabling and disabling accounts by using the Oracle Internet Directory Self-Service Console

Unlocking Accounts by Using the Oracle Internet Directory Self-Service Console If you are a member of the Security Administrators Group, then, if an account becomes locked,

you can unlock it without resetting the user password. This saves you from having to explicitly tell the user the new password. The user can simply log in by using the old password.

> **See Also:** "Unlocking User Accounts" on page 31-24 for instructions on using the Oracle Internet Directory Self-Service Console to unlock accounts

Resetting Your Own Password by Using the Oracle Internet Directory Self-Service Console

If you forget your password or become locked out of your account, then you can reset your password. This involves identifying yourself to the server by providing values for a set of password validation attributes. This takes the form of answering a password hint question to which you had earlier specified an answer.

> **See Also:** "Resetting Your Password If You Forget It" on page 31-8 for instructions on using the Oracle Internet Directory Self-Service Console to reset your password

Password Policy Error Messages

Whenever there are password policy violations, the directory server sends to the client various error and warning messages. In Oracle Internet Directory, 10*g* (9.0.4), the directory server can send these messages as LDAP controls only if the client sends a password policy request control as a part of an ldapbind or ldapcompare operation. If the client does not send the request control, then the directory server does not send the response controls. Instead, it sends errors and warnings as part of additional information.

> **See:** "Password Policy Violation Error Messages" on page H-9 for a list of the messages and information about how to resolve them

16

Directory Storage of Password Verifiers

This chapter explains how Oracle Internet Directory centrally stores password verifiers used to authenticate users to other Oracle components.

This chapter contains these topics:

About Centralized Storage of User Authentication Credentials

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. Without centralized password administration, an administrator in a large enterprise with user accounts and passwords distributed over many databases may not be able make the changes as quickly as good security requires.

Oracle Internet Directory centrally stores security credentials to make their administration easy for both end users and administrators. It 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 and Managing 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 Base64 encoded string of a one-way hashed value by 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.

Beginning with Release 9.0.4, Oracle Internet Directory stores the user password in a reversible encrypted format in an operational attribute called `orclrevpwd`. This attribute is generated only if the attribute `orclpwdencryptionenable` in the password policy entry is set to `TRUE`. The `orclrevpwd` attribute can be queried only by using the SSL one-way and two-way authentication mechanisms. This attribute cannot be queried over non-SSL sessions.

This section contains these topics:

Password Verifiers and Authentication to the Directory

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.

Hashing Schemes for Creating Password Verifiers

During installation, Oracle Universal Installer prompts you to set the one-way hashing scheme for protecting user 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.
- SSHA—Salted Secure Hash Algorithm. This is similar to SHA, but is generated by using a random salt with the password.
- SMD5—Salted MD5. This is similar to MD5, but is generated by using a random salt with the password.
- **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 manage password protection by using Oracle Directory Manager.

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

1. In the navigator pane, expand **Oracle Internet Directory Servers** and select the directory server instance for which you want to reset password hashing. The corresponding tab pages for that directory server appear in the right pane.
2. In the **System Operational Attributes** tab page, in the **Password Encryption** field, select the type of password hashing you want to use. Options are:
   - MD4
   - MD5
   - No Encryption
   - SHA
   - UNIX Crypt
   - SSHA
   - SMD5
3. Choose **Apply**.

Note: The No Encryption option specifies that user passwords are stored in clear text.

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 12-8

Storing and Managing Password Verifiers 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 for Oracle Components

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, which is then encoded as a BASE64 encoded string.

You can choose one of these hashing algorithms to derive a password verifier:

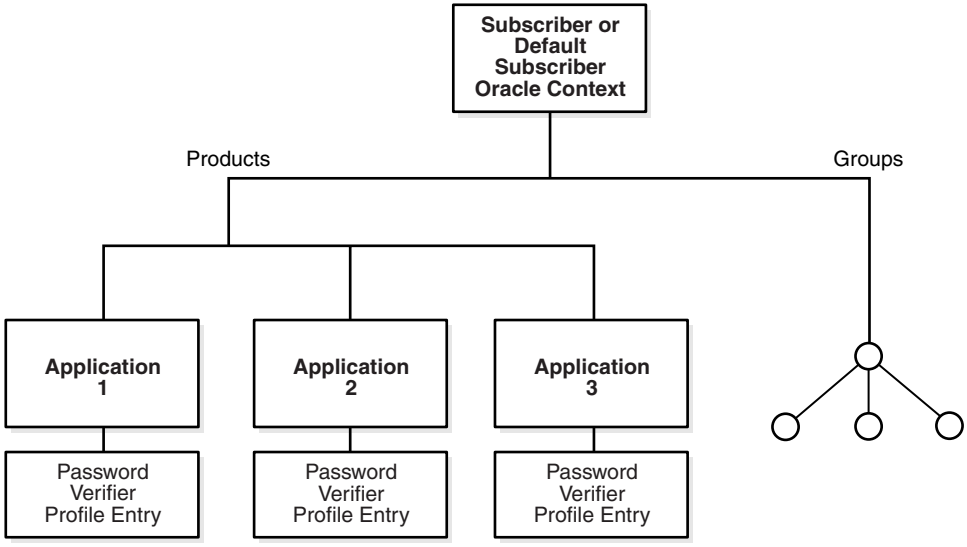
- **MD5**—An improved, and more complex, version of MD4
- **SHA**—Secure Hash Algorithm, which produces a 160-bit hash, longer than **MD5**. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.
- SSHA and SMD5
- **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 user name in the format `username@realm`.

- 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 16–1 on page 16-6: immediately below the application entry, which resides under the products entry, which, in turn, resides under the realm-specific Oracle Context.

This verifier profile entry is applicable to users in the specified realm only. For verifier generation to take effect, you must set the `orclcommonusersearchbase` attribute in the common entry of the realm-specific Oracle context to the appropriate value.

Figure 16–1 Location of the Password Verifier Profile Entry



Attributes for Storing Password Verifiers

Both the directory and Oracle components store the user password in the user entry, but in different attributes. Whereas the directory stores user passwords in the

userPassword attribute, Oracle components store user password verifiers in the authPassword, orclPasswordVerifier, or orclpassword attribute. Table 16–1 describes each of the attributes used by Oracle components.

Table 16–1 Attributes for Storing Password Verifiers in User Entries

Attribute	Description
authPassword	Attribute for storing a password to an Oracle component when that password is the same as that used to authenticate the user to the directory, namely, userpassword. The value in this attribute is synchronized with that in the userpassword attribute. 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 attribute is modified, then the authpasswords for all applications are regenerated.
orclPasswordVerifier	Attribute for storing a password to an Oracle component when that password is different from that used to authenticate the user to the directory, namely, userpassword. The value in this attribute is not synchronized with that in the userpassword attribute. Like authPassword, this attribute is multivalued and can contain all the other verifiers that different applications use for this user's clear text password.
orclPassword	Attribute for storing only the 03LOGON verifier for enterprise users. The 03LOGON verifier is synchronized with the userpassword attribute, and it is generated by default for all user entries associated with the orcluserv2 object class. When Oracle Internet Directory is installed, a database security profile entry is created by default in the Root Oracle Context. The presence of this entry triggers the generation of 03LOGON verifiers for user entries associated with the orcluserv2 object class.

Each of these attribute types has appID as an attribute subtype. This attribute subtype uniquely identifies a particular application. For example, the appID can be

the ORCLGUID of the application entry. This attribute subtype is generated during application installation.

In Figure 16–2 on page 16-9, various Oracle components store their password verifiers in Oracle Internet Directory. Oracle Application Server Single Sign-On uses the same password as that for the directory, and hence stores it in the authPassword attribute.The other applications use different passwords and hence store their verifiers in orclPasswordVerifier attribute.

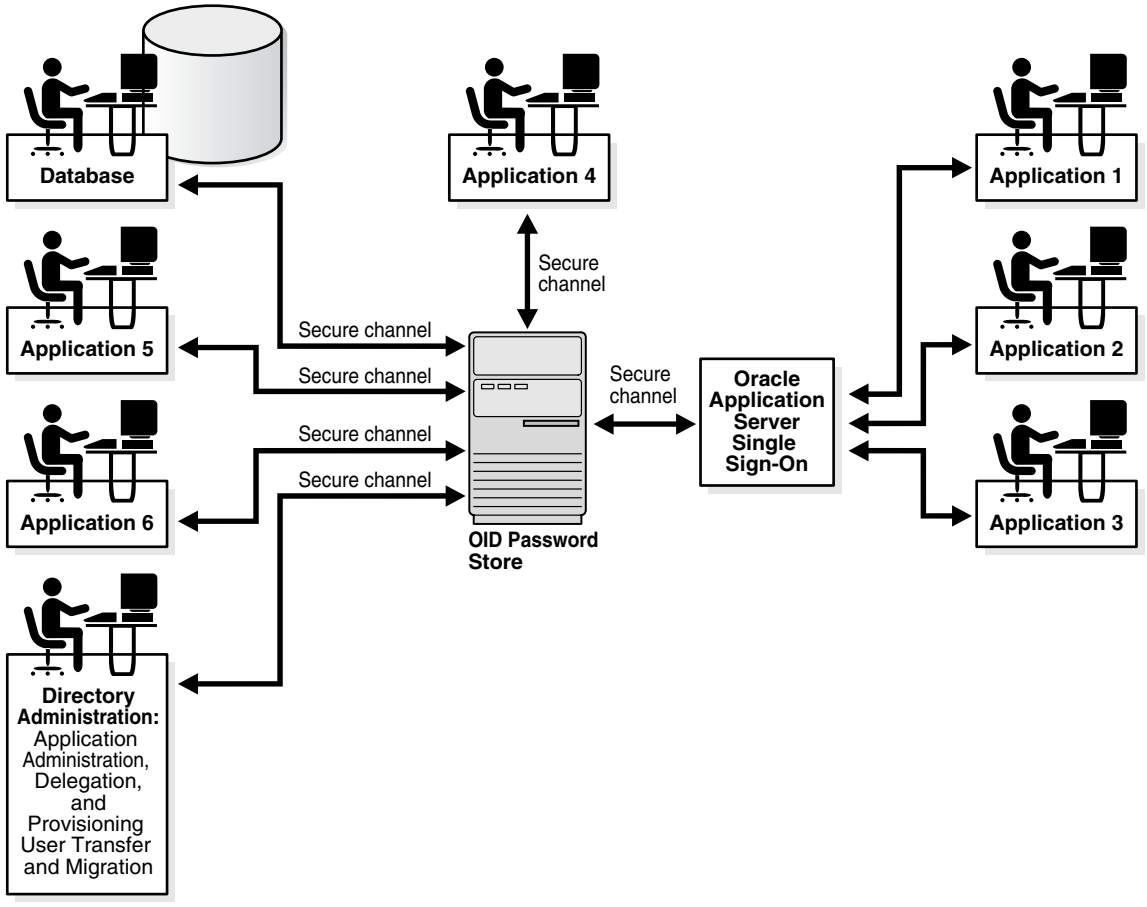
The following is an example of an application verifier profile:

```
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
orclpwdverifierparams;authpassword: crypto:ORCLWEBDAV $ realm:dc=com
    $ usernameattribute: mail
    $ usernamecase: lower
    $ nodomain: TRUE
```

SASL/MD5 and ORCLWEBDAV verifiers are generated by using user name, realm, and password. The user name attribute to be used can be specified in the verifier profile entry. The case of the user name can also be specified as either upper or lower. The ORLWEBDAV verifier is generated by appending the name of the identity management realm to the user name. If this is not required, then the verifier profile entry must specify nodomain: TRUE.

In the previous example, ORCLWEBDAV verifier is generated by using the value of the mail attribute without appending the name of the realm. Also, the user name is converted to lower case before generating the verifier.

Figure 16–2 Authentication Model



Default Verifiers for Oracle Components

To save you from having to create a profile for each Oracle component, and to enable sharing of password verifiers across all components, Oracle Internet Directory provides a default set of password verifiers. The default verifier types are MD5, MD5-IFS (SASL/MD5 with the user name set to the value of the nickname attribute and realm = Authorized_Users), WEBDAV, ORCLLM, and ORCLNT.

Two profile entries are required: one for applications using personal identification numbers (PINs), which use numeric values only, and another for applications using alphanumeric passwords.

The verifiers for PIN-based applications—for example, the voice mail application in Oracle9*i*AS Unified Messaging—are stored in the `orclpasswordverifier` attribute. The verifiers for alphanumeric password-based applications—for example, Oracle Internet File System—can be stored in either:

- The `authpassword` attribute—If an application requires its verifier to be synchronized with the `userpassword` attribute
- The `orclpasswordverifier` attribute—If synchronization with the `userpassword` attribute is not required

These profile entries also contain the list of subscribed applications and these are specified as values in the `uniquemember` attribute in the profile entries. By default, the DN of the Oracle Application Server Single Sign-On identity is one of the subscribed applications. This means that Oracle Application Server Single Sign-On is a proxy member for all its partner applications. All applications not based on Oracle Application Server Single Sign-On must add their identities (DNs) to the `uniquemember` attribute in the appropriate profile entry.

The following is an example of the profile entries.

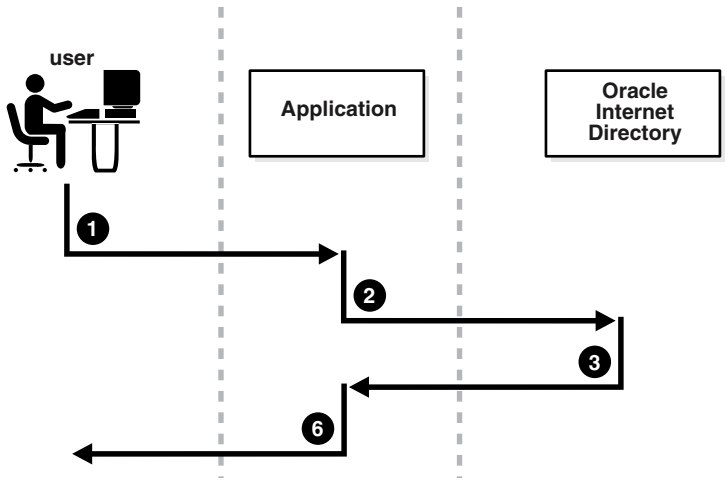
```
Cn=defaultSharedPwdProfileEntry, cn=common, cn=products, cn=oraclecontext
Objectclass: orclpwdverifierprofile
Cn: orclcommonpwdprofileentry
Orclappid: orclcommonpwd
Orclpwdverifierparams;authpassword: crypto:SASL/MD5 $ realm:Authorized_Users
Orclpwdverifierparams;authpassword: crypto:ORCLWEBDAV $ realm:Authorized_Users
Orclpwdverifierparams;authpassword: crypto:ORCLLM
Orclpwdverifierparams;authpassword: crypto:ORCLNT
Orclpwdverifierparams;orclpasswordverifier: crypto:SSHA
Uniquemember: cn=SSO,cn=Products,cn=OracleContext
Uniquemember: cn=IFS,cn=Products,cn=OracleContext

Cn=defaultSharedPINProfileEntry, cn=common, cn=products, cn=oraclecontext
Objectclass: orclpwdverifierprofile
Cn: orclcommonpinprofileentry
Orclappid: orclcommonpin
Orclpwdverifierparams;orclpasswordverifier: crypto:MD5
Orclpwdverifierparams;orclpasswordverifier: crypto:SSHA
Uniquemember: cn=SSO,cn=Products,cn=OracleContext
Uniquemember: cn=Unified Messaging,cn=Products,cn=OracleContext
```

For PIN-based applications, `authpassword` is not an option. Such applications use the `orclpasswordverifier` attribute.

Example: How Password Verification Works for an Oracle Component

Figure 16–3 shows an example of password verification for an Oracle component. In this example, the Oracle component stores its password verifiers in the directory.

Figure 16–3 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:
   a. Generates a password verifier by using the hashing algorithm specified for the particular application
   b. Compares this password verifier with the corresponding password verifiers in the directory. 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_appplication_entity" -w "password" -b
      "DN_of_the_user" -a orclpasswordverifier; appID -v password_of_the_user
      ```

   c. Notifies the application of the results of the compare operation.

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:

1. Hashes the clear text password entered by the user
2. Retrieves from the directory the hashed value of the clear text password as entered by the user
3. Initiates a challenge to the user to which the client responds. If the response is correct, then the application authenticates the user.

Managing Password Verifier Profiles for Oracle Components 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 for an Oracle Component by Using Oracle Directory Manager

To view an application's password verifiers:

1. In the navigator pane, expand in succession **Oracle Internet Directory Servers** ***directory server instance***.
2. 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
3. Choose the password verifier you want to view. This displays the Password Verifier Profile dialog box for that password verifier. The fields in this dialog box are described in Table C–12 on page C-9.
4. To modify the hashing algorithm used to generate a password verifier, in the Password Verifier Profile dialog box, enter the new value in the **Oracle Password Parameters** field.

Managing Password Verifier Profiles for Oracle Components by Using Command-Line Tools

You can view and modify 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.

Example: Modifying a Password Verifier Profile by Using Command-Line Tools

This 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
```

17

Delegation of Privileges for an Oracle Technology Deployment

This chapter explains how to store all the data for users, groups, and services in one repository, and delegate the administration of that data to various administrators. It also explains the default security configuration in Oracle Internet Directory.

This chapter contains these topics:

Delegation in the Oracle Identity Management Model

Oracle Identity Management enables you to store all the data for users, groups, and services in one repository, and to delegate a particular administrator for each set of data. By providing both a centralized repository and customized delegated access, Oracle Identity Management is both secure and scalable.

This section contains these topics:

How Delegation Works

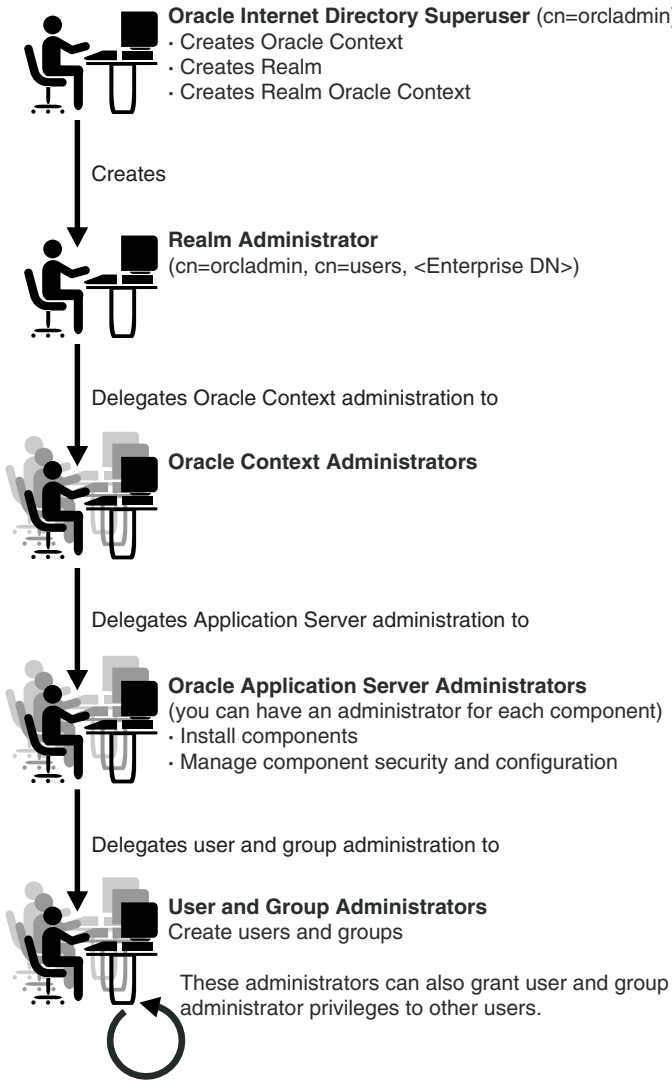
Using the delegation model, a global administrator can delegate to realm administrators the privileges to create and manage the identity management realms for hosted companies. Realm administrators can, in turn, delegate to end users and groups the privileges to change their application passwords, personal data, and preferences. Each type of user can thus be given the appropriate level of privileges.

To delegate the necessary privileges, you assign the user to the appropriate administrative group. For example, suppose that you store data for both enterprise users and the e-mail service in the directory, and need to specify a unique administrator for each set of data. To specify a user as the administrator of enterprise users, you assign that user to, say, the Enterprise User Administrators Group. To specify a user as the administrator of the e-mail services, you assign that user to, say, the E-mail Service Administrators Group.

Delegation in an Oracle Application Server Environment

Figure 17–1 shows the flow of delegation in an Oracle Application Server environment.

Figure 17–1 Delegation Flow in an Oracle Application Server Environment



As Figure 17–1 on page 17-3 shows, in an Oracle Application Server environment the directory super user creates:

- The Oracle Context
- The realm
- The realm-specific Oracle Context
- The entry for the realm administrator

The realm administrator, in turn, delegates administration of the Oracle Context to specific users by assigning those users to the Oracle Context Administrators Group. Oracle Context Administrators then delegate administration of the Oracle Application Server to one or more users by assigning them to the Oracle Application Server Administrators Group. These administrators install and administer Oracle Application Server components and delegate administration of user and group data to other administrators. The latter can, in turn, delegate others to administer user and group data.

About the Default Configuration

When you first install Oracle Internet Directory, the default configuration establishes access control policies at various points in the directory information tree (DIT). Default access controls are placed on the User and Group containers as described later in this chapter. Likewise, default privileges for specific directory entities are discussed later in this chapter. In addition, certain default privileges are granted to everyone and to each user as described in Table 17–1.

Table 17–1 Default Privileges Granted to Everyone and to Each User

Subject	Default Privileges
Everyone	The following privileges at the Root DSE: ■ Permission to browse user entries ■ Search, read, and compare access for all user attributes except the following userpkcs12, orcluserpkcs12hint, userpassword, orclpassword, and orclpasswordverifier
Each user	Complete access to his or her own attributes—including the `userpassword`, `orclpassword`, and `orclpasswordverifier` attributes.

You can customize this default configuration to meet the security requirements of your enterprise.

Overview: Privileges for Administering the Oracle Technology Stack

Administering the Oracle technology stack requires the privileges described in Table 17–2.

Table 17–2 Privileges for Administering the Oracle Technology Stack

Required Privileges	Description	More Information
User and group management privileges	These are delegated to either Oracle components that use the identity management infrastructure or to end users themselves	"Delegation of Privileges for User and Group Management" on page 17-6
Deployment-time privileges	These are required to deploy any Oracle component. They may include privileges to create appropriate entries inside the directory, or to store metadata in a common repository. Such privileges need to be given, for example, to an administrator of OracleAS Portal.	"Delegation of Privileges for Deployment of Oracle Components" on page 17-11
Runtime privileges	These are required to facilitate the runtime interactions of Oracle components within the identity management infrastructure. These include privileges to view user attributes, add new users, and modify the group membership. Such privileges need to be given to the administration tool specific to each Oracle component, enabling it to access or create entries inside Oracle Internet Directory.	"Delegation of Privileges for Component Runtime" on page 17-15

Caution: Be careful when modifying the default ACLs in any Oracle Context. Doing so can disable the security of Oracle components in your environment. See component-specific documentation for details on whether you can safely modify the default ACLs in an Oracle Context.

See Also: "Migrating an Existing Directory into the Default Directory Structure" on page 23-9 if you have an existing directory structure that you now want to migrate to an Oracle Application Server environment

Delegation of Privileges for User and Group Management

Administrative privileges are delegated to either Oracle components that use the identity management infrastructure or to end users themselves. A privilege can be delegated to either an identity—for example, a user or application—or to a role or group.

This section contains these topics:

How Privileges Are Granted for Managing User and Group Data

To delegate administrative privileges, the Oracle Internet Directory super user does the following:

1. Creates an identity management realm
2. Identifies a special user in that realm who is called the realm administrator
3. Delegates all privileges to that realm administrator

This realm administrator, in turn, delegates certain privileges that Oracle components require to the Oracle defined roles—for example, Oracle Application Server administrators. The Oracle components receive these roles when they are deployed.

In addition to delegating privileges to roles specific to Oracle components, the realm administrator can also define roles specific to the deployment—for example, a role for help desk administrators—and grant privileges to those roles. These delegated administrators can, in turn, grant these roles to end users. In fact, because a majority of user management tasks involve self-service—like changing a phone number or specifying application-specific preferences—these privileges can be delegated to end users by both the realm administrator and Oracle component administrators.

In the case of a group, one or more owners—typically end users—can be identified. If they are granted the necessary administrative privileges, then these owners can manage the group by using Oracle Internet Directory Self-Service Console, Oracle Directory Manager, or command-line tools.

Default Privileges for Managing User Data

Managing users involves privileges to:

- Create and delete user entries
- Modify user attributes
- Delegate user administration to other users

The **access control policy point** (ACP) for creating users is at the Users container in the identity management realm.

This section describes each of these privileges in more detail.

Creating Users for a Realm

To create users for a realm, an administrator must be a member of the Subscriber DAS Create User Group. Table 17–3 describes the characteristics of this group.

Table 17–3 Characteristics of the Subscriber DAS Create User Group

Characteristic	Description
Default ACP	The ACL at the Users container in the default realm allows the Subscriber DAS Create User Group in the realm Oracle Context to create users under the Users container.
Administrators	The Oracle Internet Directory super user Members of the Oracle Context Administrators Group Members of the User Privilege Assignment Group Members of the DAS Administrators Group Owners of this group
DN	`cn=oracleDASCreateUser,cn=groups,`*`Oracle_Context_DN`*.

Modifying Attributes of a User

To modify user attributes, an administrator must be a member of the Subscriber DAS Edit User Group. Table 17–4 describes the characteristics of this group.

Table 17–4 Characteristics of the Subscriber DAS Edit User Group

Characteristic	Description
Default ACP	The ACL at the Users container in the default identity management realm allows the Subscriber DAS Edit User Group in the realm Oracle Context to modify various attributes of users.

Table 17–4 (Cont.) Characteristics of the Subscriber DAS Edit User Group

Characteristic	Description
Administrators	The Oracle Internet Directory super user
	Members of the Oracle Context Administrators Group
	Members of the User Privilege Assignment Group
	Members of the DAS Administrators Group
	Owners of this group
DN	`cn=oracleDASEditUser,cn=groups,`*`Oracle_Context_DN`

Deleting a User

To delete a user in a realm, an administrator must be a member of the DAS Delete User Group. Table 17–5 describes the characteristics of this group.

Table 17–5 Characteristics of the DAS Delete User Group

Characteristic	Description
Default ACP	The ACL at the Users container in the default identity management realm allows the DAS Delete User Group in the realm Oracle Context to delete a user from the realm.
Administrators	The Oracle Internet Directory super user
	Members of the Oracle Context Administrators Group
	Members of the User Privilege Assignment Group
	Members of the DAS Administrators Group
	Owners of this group
DN	`cn=oracleDASDeleteUser,cn=groups,`*`Oracle_Context_DN`

Delegating User Administration

A delegated administrator can perform specified operations within the directory and requires permission to add any user to the User Creation, User Edit, or User Delete Groups described previously.

To grant user administration privileges to a delegate administrator, the granting administrator must be a member of the User Privilege Assignment Group. Table 17–6 describes the characteristics of this group.

Table 17–6 Characteristics of the User Privilege Assignment Group

Characteristic	Description
Default ACP	The ACL policy for each of the groups previously mentioned allows members of the User Privilege Assignment Group to add users to or remove them from those groups.
Administrators	The Oracle Internet Directory super user
	Oracle Context Administrators Group
	Owners of this group. The DNs of these owners are listed as values of the `owner` attribute in the group.
DN	`cn=oracleDASUserPriv,cn=groups,`*`Oracle_Context_DN`*

Default Privileges for Managing Group Data

Managing users and groups involves privileges to:

- Create and delete group entries
- Modify group attributes
- Delegate group administration to other users

The ACP for creating groups is at the Groups container in the identity management realm.

Creating Groups

To create groups in Oracle Internet Directory, an administrator must be a member of the Group Creation Group. Table 17–7 describes the characteristics of this group.

Table 17–7 Characteristics of the Group Creation Group

Characteristic	Description
Default ACP	The ACL at the Groups container in the realm allows the Group Creation Group to add new groups in the realm.
Administrators	The Oracle Internet Directory super user
	Members of the Oracle Context Administrators Group
	Members of the Oracle Application Server Administrators Group
	Members of the Group Privilege Assignment Group
	Members of the DAS Administrators Group
	Owners of this group

Table 17–7 (Cont.) Characteristics of the Group Creation Group

Characteristic	Description
DN	`cn=oracleDASCreateGroup,cn=groups,`*`Oracle_Context_DN`*

Modifying the Attributes of Groups

To modify the attributes of groups under the Groups container in a realm, an administrator must be a member of the Group Edit Group. Table 17–8 describes the characteristics of this group.

Table 17–8 Characteristics of the Group Edit Group

Characteristic	Description
Default ACP	The ACL at the Groups container in the realm allows the Group Edit Group to modify various attributes of groups in the realm.
Administrators	The Oracle Internet Directory super user
	Members of the Oracle Context Administrators Group
	Members of the Oracle Application Server Administrators Group
	Members of Group Privilege Assignment Group
	Members of the DAS Administrators Group
	Owners of this group
DN	`cn=oracleDASEditGroup,cn=groups,`*`Oracle_Context_DN`*

Deleting Groups

To delete groups, an administrator must have membership in the Group Delete Group. Table 17–9 describes the characteristics of this group.

Table 17–9 Characteristics of the Group Delete Group

Characteristic	Description
Default ACP	The ACL at the Groups container in the realm allows the Group Delete Group to delete groups in the realm.
Administrators	The Oracle Internet Directory super user
	Members of the Oracle Context Administrators Group
	Members of the Group Privilege Assignment Group
	Members of the DAS Administrators Group
	Owners of this group

Table 17–9 (Cont.) Characteristics of the Group Delete Group

Characteristic	Description
DN	`cn=oracleDASDeleteGroup,cn=groups,`*Oracle_Context_DN*

Delegating Group Administration

To delegate group administration to other users—that is, to add or remove users from the Group Creation, Group Edit, or Group Delete Groups described previously—an administrator must be a member of the Group Privilege Assignment Group. Table 17–10 describes the characteristics of this group.

Table 17–10 Characteristics of the Group Privilege Assignment Group

Characteristic	Description
Default ACP	The ACL policy for the Group Creation, Group Edit, or Group Delete Groups allows members of Group Privilege Assignment Group to add users to or remove them from those groups.
Administrators	The Oracle Internet Directory super user
	Members of the Oracle Context Administrators Group
	Owners of the group. The DNs of these owners are listed as values of the `owner` attribute in the group.
DN	`cn=oracleDASUserPriv,cn=groups,`*Oracle_Context_DN*

Delegation of Privileges for Deployment of Oracle Components

This section discusses the groups responsible for deploying Oracle components. It describes the tasks these administrators perform and the privileges they can grant. It includes these topics:

- How Deployment Privileges Are Granted
- Oracle Application Server Administrators
- User Management Application Administrators
- Trusted Application Administrators

Note: Oracle Internet Directory super users have all the privileges of Oracle Application Server Administrators and Trusted Application administrators, and must be members of the Oracle Application Server Administrators Group. They can:

- Assign the Oracle Application Server Administrator role to a user
- Assign the Trusted Application role to a user
- Assign the User Management Application Administrator role to a user

How Deployment Privileges Are Granted

To enable administrators to deploy Oracle components, the super user:

1. Grants certain deployment privileges to various groups—for example, the Oracle Application Server Administrators Group
2. Adds the administrators to those privileged groups

The delegated administrators, in turn, can delegate privileges to other administrators.

Oracle Application Server Administrators

Table 17–11 describes the characteristics of the Oracle Application Server Administrators Group.

Table 17–11 Characteristics of the Oracle Application Server Administrators Group

Characteristic	Description
Tasks	Perform repository database installation that creates a repository database registration entry in the directory
	Perform mid-tier installation. To associate a mid-tier with a repository, the user must have the appropriate privileges with a specific repository database.
	Install and configure Oracle Application Server components that create application entities in Oracle Internet Directory
	Grant to component entities the runtime privileges listed later in this section
	Configure provisioning profiles for components so that the components can receive update notifications
Privileges this group can delegate to components	Read Common User Attributes—except passwords, certificates, and similar security credentials
	Read common group attributes
	Create, edit, and delete groups
	Authenticate a user
	Read application verifiers
Administrators	Oracle Internet Directory super user
	Oracle Context Administrator
	Owners of this group
DN	`cn=IASAdmins,cn=groups,`*`Oracle_Context_DN`*

User Management Application Administrators

User Management Application Administrators must be members of the Oracle Application Server Administrators Group.

Table 17–12 describes the characteristics of the User Management Application Administrators Group.

Table 17–12 Characteristics of the User Management Application Administrators Group

Characteristic	Description
Tasks	User Management Application administrators install specific applications that have interfaces to perform user management operations—for example, OracleAS Portal and Oracle Application Server Wireless.
Privileges this group can delegate to components	Create, edit, and delete user attributes
Administrators	Oracle Internet Directory super user Oracle Context Administrator Owners of this group
DN	`cn=IAS & User Mgmt Admins,cn=groups,` *`Oracle_Context_DN`*

Trusted Application Administrators

Trusted Application administrators must be members of the Oracle Application Server Administrators Group.

Table 17–13 describes the characteristics of the Trusted Application Administrators Group.

Table 17–13 Characteristics of the Trusted Application Administrators Group

Characteristic	Description
Tasks	Install specific identity management components—for example, Oracle Application Server Single Sign-On, Oracle Delegated Administration Services, and Oracle Application Server Certificate Authority

Table 17–13 (Cont.) Characteristics of the Trusted Application Administrators Group

Characteristic	Description
Privileges this group can delegate to components	Read, compare, or reset the user password Proxy as the end-user Read, compare, or modify the user's certificate and SMIME certificate
Administrators	Oracle Internet Directory super user Oracle Context Administrator Owners of this group
DN	`cn=Trusted Application Admins,cn=groups,` *`Oracle_Context_DN`*

Delegation of Privileges for Component Runtime

Many Oracle components administer user entries in Oracle Internet Directory and need the corresponding privileges. For example:

- When the Oracle Application Server Single Sign-On server authenticates a user, that server:
  - Connects to Oracle Internet Directory using its own identity
  - Verifies that the password entered by the user matches that user's password stored in the directory

  To do this, the Oracle Application Server Single Sign-On server needs permission to compare user passwords. To set up the Oracle Application Server Single Sign-On cookie, it needs permission to read user attributes.
- To grant access to a user, OracleAS Portal must retrieve that user's attributes. To do this, it logs in to Oracle Internet Directory as a proxy user, impersonating the user seeking access. It therefore needs the privileges of a proxy user.

In general, Oracle components can require these privileges:

- Read and modify user passwords
- Compare user passwords
- Proxy on behalf of users accessing applications
- Administer the Oracle Context where all Oracle components store their metadata

Most Oracle components ship with a preconfigured set of privileges. You can change these default privileges to satisfy specific business requirements—for example, by removing privileges to create and delete user entries.

> **See Also:** *Oracle Application Server 10g Security Guide* for further information about the component delegation model

This section describes the security privileges required by Oracle components. It contains these topics:

- Default Privileges for Reading and Modifying User Passwords
- Default Privileges for Comparing User Passwords
- Default Privileges for Comparing Password Verifiers
- Default Privileges for Proxying on Behalf of End Users
- Default Privileges for Managing the Oracle Context
- Default Privileges for Reading Common User Attributes
- Default Privileges for Reading Common Group Attributes

Default Privileges for Reading and Modifying User Passwords

Reading and modifying user passwords requires administrative privileges on the security-related attributes in the directory—for example, the `userPassword` attribute. It requires membership in the User Security Administrators Group described in Table 17–14.

Table 17–14 Characteristics of the User Security Administrators Group

Characteristic	Description
Default ACP	The default ACL policy at the Root (DSE Entry) allows members of the User Security Administrators Group to read, write, compare, and search on `userpkcs12`, `orclpkcs12hint`, `userpassword`, `orclpassword`, and `orclpasswordverifier` attributes at the Root Oracle Context. However, directory administrators can grant similar administrative privileges to the User Security Administrators Group in the realm Oracle Context.
Administrators	The Oracle Internet Directory super user Members of the Oracle Context Administrators Group Members of the Trusted Application Administrators Group

Table 17–14 (Cont.) Characteristics of the User Security Administrators Group

Characteristic	Description
DN	`cn=oracleUserSecurityAdmins,cn=groups,` *`Oracle_Context_DN`*

Default Privileges for Comparing User Passwords

Comparing user passwords requires permission to compare a user's `userPassword` attribute. This operation is performed by components such as Oracle Unified Messaging that authenticate end users by using their passwords stored in Oracle Internet Directory.

Comparing user passwords requires membership in the Authentication Services Group described in Table 17–15.

Table 17–15 Characteristics of the Authentication Services Group

Characteristic	Description
Default ACP	The ACL policy at the Users container in the default identity management realm allows the Authentication Services Group to perform compare operation on the `userPassword` attribute of users.
Administrators	The Oracle Internet Directory super user
	Members of the Oracle Context Administrators Group
	Members of the Application Server Administrators Group
	Owners of this group
DN	`cn=authenticationServices,cn=groups,`*`Oracle_Context_DN`*

Default Privileges for Comparing Password Verifiers

To compare password verifiers, a user must have permission to compare the `userpassword` attribute. Comparing password verifiers requires membership in the Verifier Services Group described in Table 17–16.

Table 17–16 Characteristics of the Verifier Services Group

Characteristic	Description
Administrators	The Oracle Internet Directory super user
	Members of the Oracle Context Administrators group
	Members of the Application Server Administrators group
	Owners of this group
DN	`cn=verifierServices,cn=groups,`*`Oracle_Context_DN`*

Default Privileges for Proxying on Behalf of End Users

A **proxy user** has the privilege to impersonate an end user, performing on that user's behalf those operations for which that user has privileges. In an Oracle Application Server environment, the Oracle Delegated Administration Services proxies on behalf of the end user, and, through the Oracle Internet Directory Self-Service Console, performs operations on that user's behalf. In such a case, the access controls on the directory server eventually govern the operations that the user can perform.

Proxying on behalf of end users requires membership in the User Proxy Privilege Group described in Table 17–17.

Table 17–17 Characteristics of the User Proxy Privilege Group

Characteristic	Description
Default ACP	The ACL at the Users container in the default identity management realm allows User Proxy Privilege Group to proxy on behalf of the end user.
Administrators	The Oracle Internet Directory super user
	Members of the Oracle Context Administrators Group
	Owners of the groups. The DNs of these owners are listed as values of the `owner` attribute in the group or members of the Oracle Application Server Administrators Group.
	Members of the Trusted Application Administrators Group
DN	`cn=userProxyPrivilege,cn=groups,`*`OracleContextDN`*

Default Privileges for Managing the Oracle Context

To manage a specific Oracle Context, a user must have complete access to it. Managing an Oracle Context requires membership in the Oracle Context

Administrators Group described in Table 17–18. An Oracle Context Administrators Group exists for each Oracle Context and has administrative permission in the specific Oracle Context.

Table 17–18 Characteristics of the Oracle Context Administrators Group

Characteristic	Description
Default ACP	The ACL policy at the root node of the Oracle Context allows members of Oracle Context Administrators Group to perform all administrative operations within the Oracle Context. Such a policy is set up when a new Oracle Context is created in the directory.
Administrators	The Oracle Internet Directory super user Members of the Oracle Context Administrators Group
DN	`cn=oracleContextAdmins,cn=groups,`*`Oracle_Context_DN`

Default Privileges for Reading Common User Attributes

Common user attributes are: `mail`, `orclguid`, `displayname`, `preferredlanguage`, `orcltime`, `gender`, `dateofbirth`, `telephonenumber`, `wirelessaccountnumber`. To read these attributes requires membership in the Common User Attributes Group described in Table 17–19.

Table 17–19 Characteristics of the Common User Attributes Group

Characteristic	Description
Default ACP	The default ACL is on the User container in the realm and grants permission to read common user attributes.
Administrators	The Oracle Internet Directory super user Members of the Application Server Administrators Group Owners of this group
DN	`cn=commonuserattributes,cn=users,`*`Oracle_Context_DN`

Default Privileges for Reading Common Group Attributes

Common group attributes are: `cn`, `uniquemember`, `displayname`, and `description`. To read these attributes requires membership in the Common Group Attributes Group described in Table 17–20 on page 17-20.

Table 17–20 Characteristics of the Common Group Attributes Group

Characteristic	Description
Default ACP	The default ACL is on the Group container in the realm and grants permission to read these attributes: `cn`, `uniquemember`, `displayname`, and `description`.
Administrators	The Oracle Internet Directory super user Members of the Application Server Administrators Group Owners of this group
DN	`cn=commongroupattributes,cn=groups,`*`Oracle_Context_DN`*

Part IV

Directory Deployment

This part discusses important deployment considerations. It includes these chapters:

18

Directory Deployment Considerations

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

This chapter contains these topics:

- The Expanding Role of Directories
- Logical Organization Of Directory Information
- Physical Distribution: Partitions, Replicas, and High Availability
- The Oracle Directory Integration and Provisioning Platform
- Capacity Planning, Sizing, and Tuning
- Multiple installations of Oracle Internet Directory on one host

See Also:

- Chapter 20, "Capacity Planning for the Directory" for more detailed information about capacity planning
- Chapter 26, "High Availability And Failover Considerations" for more detailed information about high availability
- Chapter 21, "Tuning Considerations for the Directory" for more detailed information about tuning
- "Directory Replication and High Availability" 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

Oracle Internet Directory serves as a shared repository for the entire Oracle Identity Management infrastructure. A carefully planned logical structure of the directory enables:

- Enforcement of security policies that meet the requirements of your deployment
- A more efficient physical deployment of the directory service
- Easier configuration of synchronization of a third-party directory with Oracle Internet Directory

See Also: "Planning the Directory Information Tree for Identity Management" on page 19-5

Physical Distribution: Partitions, Replicas, and High Availability

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 them by using knowledge references

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

This section contains these topics:

An Ideal Deployment

Although it would be simpler and more secure to store all naming contexts in one central directory, this central directory would then be a single point of failure.

One 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, because multimaster replication allows "update anywhere" configurations, it is more efficient and less costly to consolidate the directory rather than to maintain 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:** "High Availability Considerations" on page 18-6 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 **Oracle9i Advanced Replication**.

- 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.

  See Also: The section on planning the physical deployment of Oracle Internet Directory in *Oracle Identity Management Concepts and Deployment Planning Guide* for more information about replicated directory configurations

High Availability 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.

- 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.

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 26, "High Availability And Failover Considerations" for further details about high-availability and failover options available with Oracle Internet Directory
- The section on planning the physical deployment of Oracle Internet Directory in *Oracle Identity Management Concepts and Deployment Planning Guide* for more information about high availability for the directory

The Oracle Directory Integration and Provisioning Platform

You can reduce administrative time and costs by integrating your applications and directories—including third-party LDAP directories—with Oracle Internet Directory. The Oracle Directory Integration and Provisioning platform, a component of Oracle Identity Management, enables you to do this. For example, you might need to do the following:

- Keep employee records in Oracle Human Resources consistent with those in Oracle Internet Directory. The Oracle Directory Integration and Provisioning platform provides this synchronization through the Oracle Directory Synchronization Service.
- Notify certain LDAP-enabled applications—such as OracleAS Portal—whenever changes are applied to Oracle Internet Directory. The Oracle Directory Integration and Provisioning platform provides this notification through the Oracle Directory Provisioning Integration Service.

Throughout the integration process, the Oracle Directory Integration and Provisioning platform ensures that the applications and other directories receive and provide the necessary information in a reliable way.

You can integrate with various directories, including Microsoft Active Directory and SunONE Directory Server. For example, in an Oracle Application Server environment, where access to Oracle components relies on data stored in Oracle Internet Directory, you can still use Microsoft Active Directory as the central enterprise directory. Users of that directory can still access Oracle components because the Oracle Directory Integration and Provisioning platform can synchronize the data in Microsoft Active Directory with that in Oracle Internet Directory.

See Also: Chapter 32, "Oracle Directory Integration and Provisioning Platform Concepts and Components"

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, Table 18–1 indicates the approximate level of CPU power required for various deployment scenarios for Oracle Internet Directory:

Table 18–1 CPU Power for Various Deployment Scenarios

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. Table 18–2gives the approximate disk space requirements for variously sized DITs.

Table 18–2 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. Table 18–3 on page 18-11 provides estimates of the memory requirements for various DIT sizes:

Table 18–3 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 20, "Capacity Planning for the Directory."

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 21, "Tuning Considerations for the Directory" for further details on various tuning tips and techniques

19

Deployment of Oracle Identity Management Realms

This chapter discusses identity management realms and how to plan and configure them for both enterprise and hosted deployments.

This chapter contains these topics:

Identity Management Realms in an Enterprise Deployment

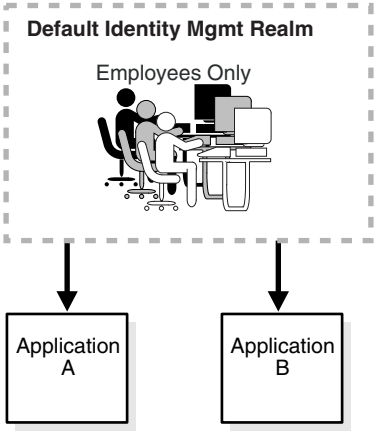
This section discusses deployments with single identity management realms and those with multiple ones. It contains these topics:

- Single Identity Management Realm in the Enterprise
- Multiple Identity Management Realms in the Enterprise

Single Identity Management Realm in the Enterprise

This is the default configuration of all Oracle products. In this case, an enterprise has a single set of users, all of whom are managed with the same identity management policies. There is only one default identity management realm in Oracle Internet Directory. All Oracle components in the enterprise serve users in the default realm. Figure 19–1 illustrates this usage.

Figure 19–1 Enterprise Use Case: Single Identity Management Realm



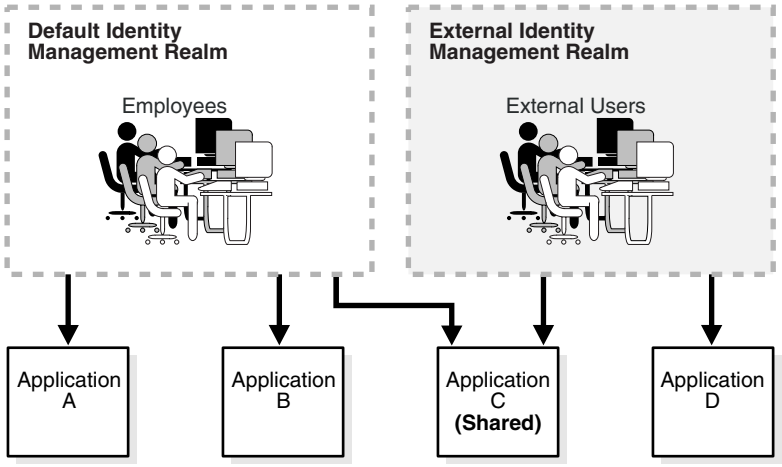
In the example in Figure 19–1, there is a single identity management realm in which all users and groups are managed and share access to the same applications.

Multiple Identity Management Realms in the Enterprise

Certain enterprises can use the same identity management infrastructure to serve both internal as well as external, self-registered users. Because the identity management policies for internal and external users are different, the enterprise can

deploy two realms, one for internal and one for external users. Figure 19–2 on page 19-3 illustrates this usage.

Figure 19–2 Enterprise Use Case: Multiple Identity Management Realms



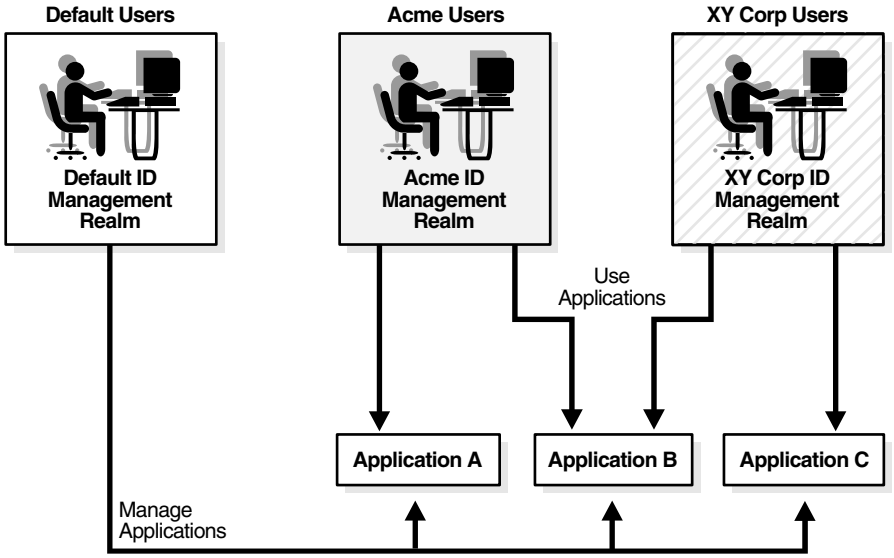
In the example in Figure 19–2, the default identity management realm is for internal users—namely, employees—and these have access to Applications A, B, and C. The external identity management realm is for external users, and they have access to Applications C and D.

Identity Management Realms in a Hosted Deployment

In a hosted deployment, the application service provider (ASP) supplies one or more companies with identity management services and hosts applications for them. Each hosted company is associated with a separate identity management realm where users of that company are managed. Users belonging to the application service provider are managed in a different realm, typically the default realm.

Figure 19–3 shows a hosted deployment with two hosted companies.

Figure 19–3 Hosted Deployment Use Case



In the example in Figure 19–3, the ASP users manage various applications hosted for the hosted company. Each hosted company has an associated identity management realm where the ASP manages its users, groups and associated policies.

Identity Management Realm Implementation in Oracle Internet Directory

Table 19–1 describes the information model in the Oracle Internet Directory tree for an identity management realm.

Table 19–1 Oracle Identity Management Objects

Object	Description
Root Oracle Context	Contains a pointer to the default identity management realm in the infrastructure. It also contains information on how to locate an identity management realm given a simple name of the realm.
Identity Management Realm	A normal directory entry in the Oracle Internet Directory tree with a special object class associated with it.

Table 19–1 (Cont.) Oracle Identity Management Objects

Object	Description
Identity Management Realm-Specific Oracle Context	In each realm, the container of the following information: ■ User naming policy of the identity management realm—that is, how users are named and located ■ Mandatory authentication attributes ■ Location of groups in the identity management realm ■ Privilege assignments for the identity management realm—for example: who has privileges to add more users to the realm. ■ Application specific data for that realm including authorizations

Planning the Directory Information Tree for Identity Management

Oracle Internet Directory serves as a shared repository for the entire Oracle Identity Management infrastructure. A carefully planned logical structure of the directory enables:

- Enforcement of security policies that meet the requirements of your deployment
- A more efficient physical deployment of the directory service
- Easier configuration of synchronization of a third-party directory with Oracle Internet Directory

Planning the logical organization of the directory for Oracle Identity Management comprises:

- Planning the overall structure of the directory information tree (DIT)
- Planning the directory containment and naming for users and groups
- Planning the identity management realm

Figure 19–4 shows the impact of each of these steps in the directory information tree.

Figure 19–4 Planning the Directory Information Tree

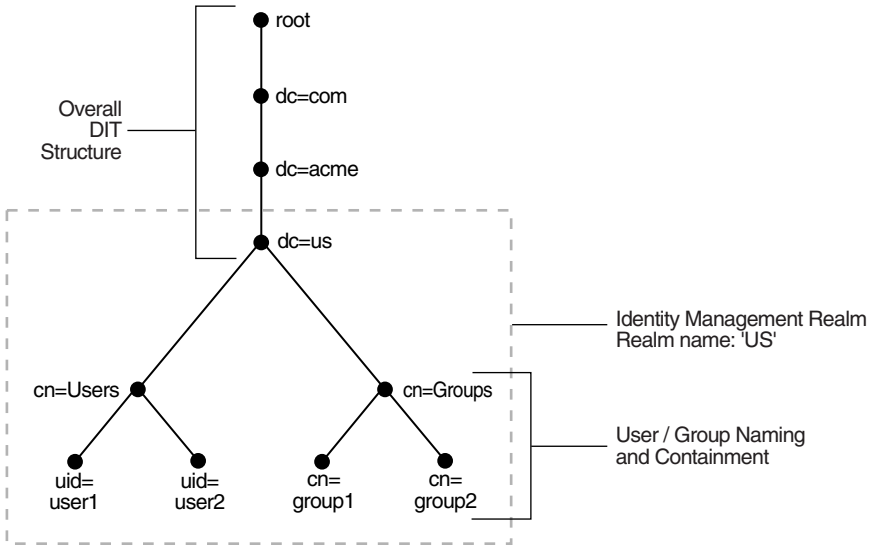


Figure 19–4 illustrates a hypothetical company, called Acme, that makes the following decisions with respect to the logical organization of the directory in their U.S. deployment:

- A domain name-based scheme is to represent the overall DIT hierarchy. Because the identity management infrastructure is being rolled out in the `us` domain, the root of the DIT is `dc=us,dc=acme,dc=com`.
- Within the naming context chosen, all users are represented under a container called `cn=users`. Within this container, all users are represented at the same level—that is, there is no organization-based hierarchy. In addition, the `uid` attribute is chosen as the unique identifier for all users.
- Within the naming context chosen, all enterprise groups are represented under a container called `cn=groups`. Within this container, all enterprise groups are represented at the same level. The naming attribute for all group entries is `cn`.
- Finally, the container `dc=us` is chosen as the root of the identity management realm. In this case, the name of the realm is `us`. The deployment expects to

enforce similar security policies for all users who fall under the scope of the `us` realm.

This section discusses further details to consider when designing the logical organization of directory information. It contains these topics:

- Planning the Overall Directory Structure
- Planning the Names and Containment of Users and Groups
- Planning the Identity Management Realm

Planning the Overall Directory Structure

This task involves designing the basic directory information tree that all identity management-integrated applications in the enterprise are to use. As you do this, keep these considerations in mind:

- The directory organization should facilitate clean and effective access control. If replication—either full or partial—is planned, then proper boundaries and policies for directory replication can be enforced only if the design of the DIT brings out the separation.
- If the enterprise is integrating with a third-party directory server, then it is best to align the DIT design of Oracle Internet Directory with the existing DIT. This consideration also applies to deployments that are rolling out Oracle Internet Directory now but plan to roll out another directory later—for example, Microsoft Active Directory that is required for the operation of software from Microsoft. In each case, choosing an Oracle Internet Directory DIT design that is more consistent with that of the third-party directory makes management of user and group objects easier through Oracle Delegated Administration Services and other middle-tier applications.
- In a single enterprise scenario, choosing a DIT design that aligns with the DNS domain name of the enterprise suffices. For example, if Oracle Internet Directory is set up in a company having the domain name `acme.com`, then a directory structure that has `dc=acme,dc=com` is recommended. Oracle Corporation recommends that you not use departmental or organization level domain components such as `engineering` in `engineering.acme.com`.
- If the enterprise has an X.500 directory service, and no other third-party LDAP directories in production, then it may benefit by choosing a country-based DIT design. For example, a DIT design with the root of `o=acme, c=US` might be more suitable for enterprises which already have an X.500 directory service.

- Because the directory can be used by several applications—both from Oracle and from third-parties alike—the naming attributes used in relative distinguished names (RDNs) constituting the overall DIT structure should be restricted to well-known attributes. The following attributes are generally well-known among most directory-enabled applications:
  - `c`: The name of a country
  - `dc`: A component of a DNS domain name
  - `l`: The name of a locality, such as a city, county or other geographic region
  - `o`: The name of an organization
  - `ou`: The name of an organizational unit
  - `st`: The name of a state or province
- A common mistake is to design the DIT to reflect either the corporate divisional or organizational structure. Because most corporations undergo frequent reorganization and divisional restructuring, this is not advisable. It is important to insulate the corporate directory from organizational changes as much as possible.

Planning the Names and Containment of Users and Groups

Most of the design considerations that are applicable to the overall DIT design are also applicable to the naming and containment of users and groups. This section offers some additional things to consider when modeling users and groups in Oracle Internet Directory.

Considerations for Users

The Oracle Identity Management infrastructure uses Oracle Internet Directory as the repository for all user identities. Even though a user might have account access to multiple applications in the enterprise, there is only one entry in Oracle Internet Directory representing that user's identity. The location and content of these entries in the overall DIT must be planned before deploying Oracle Internet Directory and other components of the Oracle Identity Management infrastructure.

- As mentioned in the previous section, it is tempting to organize users according to their current departmental affiliations and hierarchy. However, 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 that person's directory entry.

- There are no performance benefits derived from organizing users in a hierarchy according to organizational affiliations or management chain. Oracle Corporation recommends that you keep the DIT containing users as flat as possible.
- If the deployment has different user populations with each one maintained and managed by a different organization, then Oracle Corporation recommends subdividing users into containers based on these administrative boundaries. This simplifies the setting of access controls and helps in cases where replication is needed.
- The out-of-the-box default nickname attribute for uniquely identifying users in lookup operations is `uid`. This is the default attribute used for logins. The out-of-the-box default naming attribute for constructing a DN is `cn`.
- 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.
- It is required that all user entries created in the directory belong to the following object classes: `inetOrgPerson`, `orclUserV2`.
- If you already have a third-party directory, or plan to integrate with one in the future, then it is beneficial to align the user naming and directory containment subsequent administration of the distributed directories.

Note: In Oracle Internet Directory Release 9.0.2, the default value for the `nickname` attribute was `cn`. As of Release 9.0.4, the default value for this attribute is `uid`.

Considerations for Groups

Some applications integrated with the Oracle Identity Management infrastructure can also base their authorizations on enterprise-wide groups created by the deployment in Oracle Internet Directory. Like user entries, the location and content of these group entries should also be carefully planned. When you design groups, consider the following:

- There are no performance benefits to be gained from organizing enterprise groups in a hierarchy based on the organizational affiliations or ownership. Oracle Corporation recommends keeping the DIT that contains groups as flat as possible. This facilitates easy discovery of groups by all applications and fosters sharing of these groups across applications.

- It is preferable to separate the users and groups in the DIT so that different management policies can be applied to each set of entries.
- The attribute used to uniquely identify a group should be `cn` or `CommonName`.
- All group entries created by the enterprise in the directory should belong to the following object classes: `groupOfUniqueNames` and `orclGroup`. The former object class is an internet standard for representing groups. The latter is useful when using the Oracle Internet Directory Self-Service Console to manage groups.
- Instead of creating new directory access controls for each enterprise-wide group, consider doing the following:
  1. Use the `owner` attribute of the group to list which users own this group.
  2. Create an access control policy at a higher level that grants all users listed in the `owner` attribute special privileges to perform the various operations.
- In the `description` attribute, provide information for users to understand the purpose of the group.
- Consider using the `displayName` attribute from the `orclGroup` object class. This enables Oracle Delegated Administration Services and Oracle Internet Directory Self-Service Console to display a more readable name for the group.
- If you have different sets of groups, each of which is maintained and managed by a different organization with its own administrative policies, then sub-divide the groups into containers based on these administrative boundaries. This simplifies the setting of access controls. It also helps when replication is needed.
- If you already have a third-party directory, or plan to integrate with one in the future, then align the group naming and directory containment in Oracle Internet Directory with the one used in the third-party directory. This simplifies the synchronization and subsequent administration of the distributed directories.

Planning the Identity Management Realm

The previous sections describe guidelines for you to structure the overall DIT and the placement of users and groups for your deployment. Because implementing these guidelines can lead to an infinite number of deployment configurations, you need to capture the intent of your deployment in metadata in the directory itself. This metadata enables Oracle software and other third-party software relying on the Oracle Identity Management infrastructure to understand the deployment intent and successfully function in customized environments.

In Oracle Internet Directory, this deployment intent is captured in the identity management realm. The realm also helps set identity management policies for users and groups whose placement is described in the previous section.

The identity management realm is a well-scoped area in the directory that consists of:

- A well-scoped collection of enterprise identities—for example, all employees in the US)
- A collection of identity management policies associated with these identities
- A collection of groups—that is, aggregations of identities—that makes it easier to set identity management policies

Once you have decided on the overall DIT structure and the placement of users and groups, you need to identify the directory entry to serve as the root of the identity management realm. This entry determines the scope of the identity management policies defined in the realm. By default, the scope is the entire directory subtree under the root of the identity management realm. Under this entry, a special entry called `OracleContext` is created. It contains the following:

- The deployment-specific DIT design, including user and group naming and placement, as described in previous sections
- The identity management policies associated with this realm
- Additional realm-specific information specific to Oracle applications

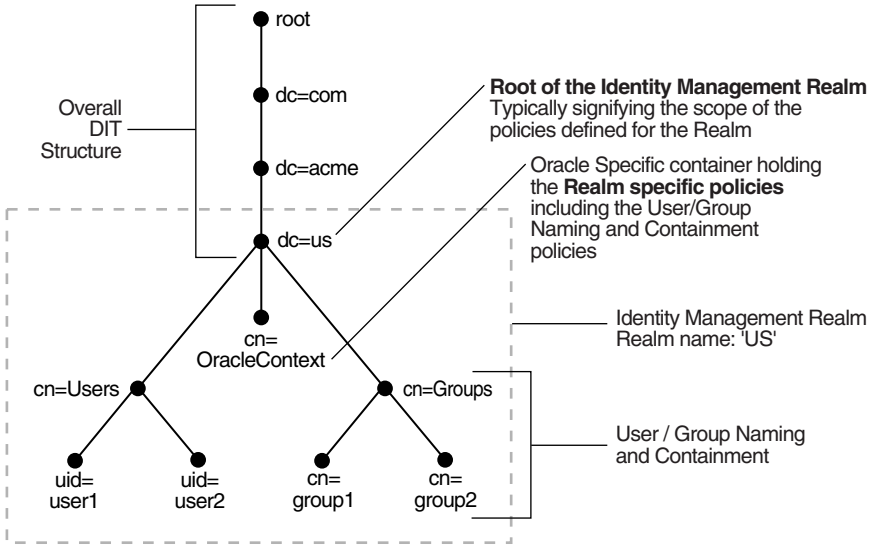
When planning the identity management realm, consider the following:

- The security needs of your enterprise must dictate the choice of the root of the identity management realm. Typically, most enterprises need only one realm. However, multiple realms may be required when multiple user populations are managed with different identity management policies.
- If you already have a third-party directory, or plan to integrate with one in the future, then align the choice of the identity management realm root with the DIT design of the third-party directory. This simplifies the synchronization and subsequent administration of the distributed directories.
- To configure and administer identity management realms, use the administrative tools provided by Oracle Internet Directory. These include the Oracle Internet Directory Configuration Assistant, the Oracle Internet Directory Self-Service Console, and command-line tools.
- Once you have used the Oracle Internet Directory tools to configure the identity management realm, plan on updating the directory naming and containment

policies to reflect the customizations made by the deployment. This update must happen prior to installing and using other Oracle components that use the Oracle Identity Management infrastructure.

Figure 19–5 shows an example of an identity management realm for an enterprise called Acme.

Figure 19–5 Example of an Identity Management Realm



In the example in Figure 19–5, the deployment has chosen to use a domain name-based DIT structure. In this case, the container `dc=us,dc=acme,dc=com` is chosen as the root of the identity management realm. This results in the creation of a new identity management realm whose scope, by default, is restricted to the entire directory subtree under the entry `dc=us`. The name of the identity management realm is `US`.

Default Directory Information Tree and Identity Management Realm

To make configuration easier, Oracle Internet Directory, at installation, creates a default DIT and sets up a default identity management realm. It gives you the option of setting up a DIT based on the domain of the computer on which the installation is performed. For example, if the installation is on a computer named `oidhost.us.mycompany.com`, then the root of the default identity management

realm is dc=us,dc=mycompany,dc=com. Oracle Internet Directory creates the following:

- An Oracle Context associated with the default identity management realm. The Oracle Context stores all the realm-specific policies and metadata. Using the example in the previous paragraph, it creates the Oracle Context with the distinguished name cn=OracleContext,dc=us,dc=mycompany,dc=com. This entry and the nodes under it enable Oracle software to detect realm-specific policies and settings.
- A directory structure and naming policies in the default identity management realm. These enable Oracle components to locate various identities. The default values for these are:
  - All users are located in the container cn=users under the base of the identity management realm. In this example, it is cn=users,dc=us,dc=mycompany,dc=com.
  - Any new users created in the identity management realm using the Oracle Identity Management infrastructure are also created under the cn=users container.
  - All new users created in the identity management realm using the Oracle Identity Management infrastructure belong to the object classes orclUserV2 and inetOrgPerson.
  - All groups are located in the container cn=groups under the base of the identity management realm. In this example, it is cn=groups,dc=us,dc=mycompany,dc=com.
- Identity management realm administrator. This user, called cn=orcladmin, is located under the users container. In this example, the fully qualified DN of the bootstrap user is cn=orcladmin,cn=users,dc=us,dc=mycompany,dc=com.
- Default authentication policies, which enable authentication services to perform the appropriate actions. These include:
  - The default directory password policy—for example, password length, lockout, and expiration
  - Additional password verifiers that need to be automatically generated when provisioning the user
- Identity management authorizations. Oracle Internet Directory grants these to the bootstrap user who can further delegate these authorizations through the

Oracle Internet Directory Self-Service Console. Some of these authorizations include:

- Common identity management operational privileges—for example, user creation, user profile modification, group creation
- Privileges to install new Oracle components by using the Oracle Identity Management infrastructure.
- Privileges to administer Oracle Internet Directory Self-Service Console

See Also:

- "Optional Attributes of the orclUserV2 Object Class" on page 17 for more information about the `orclUserV2` object class
- Chapter 17, "Delegation of Privileges for an Oracle Technology Deployment" for a fuller description of the default access control policies in Oracle Identity Management

Administration of Identity Management Realms

This section describes the various administrative tasks that you can perform with respect to identity management realms. It contains these topics:

- Customizing an Existing Identity Management Realm
- Creating Additional Identity Management Realms

Customizing an Existing Identity Management Realm

Once a realm is created, you can further customize various aspects of it. Table 19–2 lists the aspects you can customize, the tools available for each type of customization, and where to look for more information.

Table 19–2 Customizing an Existing Realm

What You Can Customize	Tools	Information
Directory structure and naming policies	Oracle Delegated Administration Services Oracle Directory Manager Command-line tools	"Planning the Directory Information Tree for Identity Management" on page 19-5 "Using the Oracle Internet Directory Self-Service Console" on page 31-4
Authentication policies	Oracle Directory Manager Command-line tools	Chapter 15, "Password Policies in Oracle Internet Directory"
Identity management authorizations	Oracle Delegated Administration Services Oracle Directory Manager Command-line tools	Chapter 17, "Delegation of Privileges for an Oracle Technology Deployment" "Using the Oracle Internet Directory Self-Service Console" on page 31-4

Note: Be sure to complete all of the customizations before installing any Oracle component that uses the realm. Changing the properties of a realm after an Oracle component has been deployed can result in unpredictable behavior.

See Also: If you are integrating with Oracle Application Server Single Sign-On, the section "Updating the Single Sign-On Server with Directory Changes" in the *Oracle Application Server Single Sign-On Administrator's Guide*

Creating Additional Identity Management Realms

You can create additional identity management realms by using the Oracle Internet Directory Self-Service Console.

Note: Not all applications can work with multiple identity management realms

Whenever you add an additional realm, you may need to make existing applications aware of it by using a manual procedure. For more information, see the application-specific documentation.

In the Oracle Identity Management infrastructure, the single sign-on server needs to be made aware of an additional realm by using a special administrative procedure. Please refer to the chapter "Single Sign-On in Multiple Realms" in the *Oracle Application Server Single Sign-On Administrator's Guide* for instructions on enabling multiple realms in Oracle Application Server Single Sign-On.

See Also: "Creating an Additional Identity Management Realm by Using the Oracle Internet Directory Self-Service Console" on page 31-14

20

Capacity Planning for the Directory

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 e-mail 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 e-mail 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 every 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 e-mail messaging software employed by our hypothetical company, Acme Corporation. The e-mail 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 e-mail 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:

Table 20–1 Assumptions about Entry Types and Their Sizes

Entry Type	Size
Total user population	40,000
Average number of private aliases for each person	10
Average number of private distribution lists for each 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 these assumptions, we can derive the overall count of entries in Oracle Internet Directory as:

Table 20–2 Overall Count of Entries

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

These 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 e-mails everyday and receives an average of 10 e-mails a day from the outside world. Assuming that there are, on an average, five recipients for each e-mail being sent by a user, this would result in five directory lookups for each e-mail.

The following table summarizes all the possible directory lookups that can happen in one day:

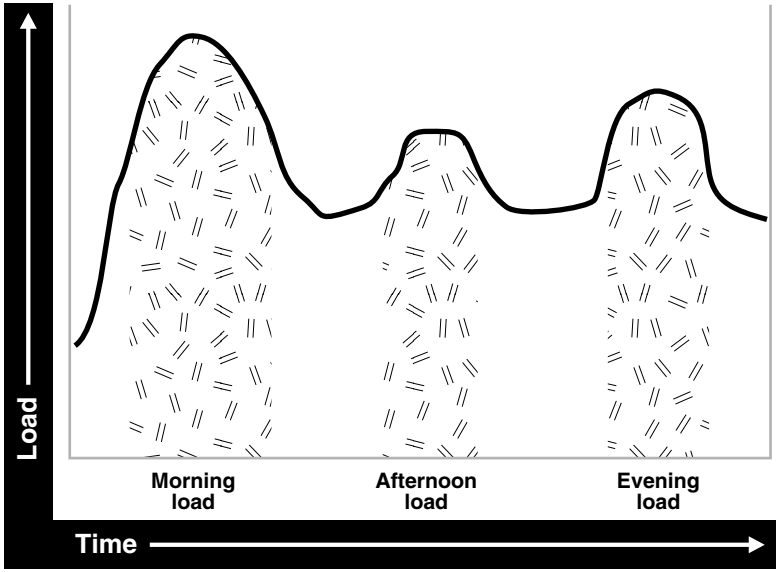
Table 20–3 Directory Lookups in a Single Day

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, and so on)	800,000

Summing up, the total number of directory lookups everyday would be about 3,200,000 (3.2 million) directory lookups everyday. If these directory lookups were spread out uniformly along the day, it would require about 37 directory lookups every second (133,333 lookups every hour). Unfortunately, we will never have this case.

Usage analysis of the current e-mail system over a period of 24 hours shows the pattern illustrated in Figure 20–1.

Figure 20–1 Usage Analysis of Current E-mail System



The e-mail 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 for an 8 hour workday:

Table 20–4 Working Hour Loads

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 every hour)
Afternoon load	10%: 0.90 x 0.10 x 3,200,000 = 288,000 lookups for 1 hour (288,000 lookups every hour)
Evening load	20%: 0.90 x 0.20 x 3,200,000 = 576,000 lookups for 2 hours (288,000 lookups every hour)

These calculations indicate that the Oracle Internet Directory in this case should be designed to handle the peak load of 936,000 lookups every 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 21, "Tuning Considerations for the Directory". 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:

Table 20–5 Disk Space Requirements

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

The data shown in the previous table makes the following assumptions:

- There are about 20 cataloged attributes.
- There are about 25 attributes for each 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 for each 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:

Table 20–6 Tablespaces Used to Store Oracle Internet Directory Data

Tablespace Name	Contents
`OLTS_ATTR_STORE`	Stores all of the attributes for all entries in the DIT
`OLTS_CT_STORE`	Stores all the remaining (including user-defined) catalogs and the indexes defined in the catalogs
`OLTS_DEFAULT`	Stores all of the data pertaining to the administration of the Oracle Internet Directory as well as the data used for replication support
`OLTS_SVRMGSTORE`	Stores all the tables and indexes required for Oracle Internet Directory Server Manageability
`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 earlier. All of the size calculations are based on the following variables:

Table 20–7 Variables Used for Size Calculation

Variable Name	Description
`num_entries`	Total number of entries in the directory
`attrs_per_entry`	Average number of attributes for each 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

Table 20–7 (Cont.) Variables Used for Size Calculation

Variable Name	Description
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 for each 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 indexes. 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
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 for each statistics entry

Using the variables shown in Table 20–7, the size of individual tablespaces can be calculated as follows:

Table 20–8 Size of Individual Tablespaces

Tablespaces Containing Tables	Formula
ATTRSTORE_INDEX_SIZE	num_entries*(attrs_per_entry+6) *10
CATALOG_INDEX_SIZE	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
CN_SIZE	num_entries*avg_dn_size*db_storage_ovhd
DN_INDEX_SIZE	num_entries*2*(avg_dn_size * 3)
DN_SIZE	num_entries*2*(avg_dn_size+4)
OBJECTCLASSES_SIZE	num_entries*objectclass_per_entry*objectclass_size*db_storage_ovhd + num_auditlog_entries*2*avg_dn_size*db_storage_ovhd
OLTS_ATTR_STORE	(num_entries*(((attrs_per_entry)*(avg_attrname_len+avg_attr_size+22))+6*35)*db_storage_ovhd)+attrstore_index_size
OLTS_BATTRSTORE	6M+(((num_binary_attrs*avg_binval_length)+6*35)*db_storage_ovhd)
OLTS_CT_STORE	(cn_size+objectclasses_size+dn_size+catalog_index_size+dn_index_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_SVRMGSTORE	2M+num_stats_entries*((avg_attrname_len+avg_attr_size+20)*(2*attrs_per_stats_entry)*db_storage_ovhd*(orclstatsperiodicity/10)*12)
SYSTEM	300MB

Use the arithmetic operations shown in the preceding table to 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.

Table 20–9 Values for Variables Used for Sizing Calculations

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 for each user)
num_acis	80,000 ACIs (2 for each user)
num_auditlog_entries	1000
db_storage_ovhd	1.4 (40% overhead)
db_index_ovhd	5.0 (400% overhead)
factor_of_safety	1.5 (50% factor of safety)
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:

Table 20–10 Tablespace Sizes

Tablespace Name	Size in Bytes	Size in MB
OLTS_ATTRSTORE	2,223,000,000	2182
OLTS_CT_STORE	2,328,512,000	274
OLTS_DEFAULT	159,680,000	156

Table 20–10 (Cont.) Tablespace Sizes

Tablespace Name	Size in Bytes	Size in MB
OLTS_SVRMGSTORE	2,701,568	3
SYSTEM	314572800	300
Total Size	**5038093862**	**4920**

The previous table 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, then the CPUs work less efficiently and system performance drops. Furthermore, memory usage increases in proportion to the number of concurrent connections to the database and the number of concurrent users of the directory. For the purposes of capacity planning, an active connection begins when a client seeks to bind to the directory and ends when that bind is terminated.

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 memory usage for each process 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 * [150+ {attrs_per_entry + 6) * (avg_attrname_len + avg_attr_size + 40) }] * 1.3

> **See Also:** Chapter 21, "Tuning Considerations for the Directory" for further information on SGA tuning

The following table gives minimum memory requirements for different directory configurations:

Table 20–11 Minimum Memory Requirements for Different Directory Configurations

Directory Type	Entry Count	Minimum Memory
Small	Less than 600,000	512 MB
Medium	600,000 to 2,000,000	1 GB
Large	Greater than 2,000,000	2 GB

Going back to our example of Acme Corporation, the number of entries in the directory are close to 1,000,000 (1 million). Oracle Corporation recommends choosing the 2 GB option in order to maximize performance.

Network Requirements

The network is rarely a bottleneck in most installations. However serious consideration must be given to it during the capacity planning stage. If the clients do not get adequate network bandwidth to send and receive messages from Oracle Internet Directory, the overall throughput will seem to be very low. For example, if we have configured Oracle Internet Directory to service 800 search operations every 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 every 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:

Table 20–12 Maximum Possible Throughput for Two Types of Operations

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 every hour which results in an equivalent number of lookup operations to the directory. This requires about 260 directory operations every 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:

- Detailed Calculations of CPU Requirements

CPU Configuration

The CPU sizing for Oracle Internet Directory is directly a function of the user workload. The following factors will determine CPU configuration:

- The number of concurrent operations you want to support. This will be directly dependent on the number of users performing operations simultaneously.
- The acceptable latency of each operation. For example, in an e-mail application, a latency for each 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:

Table 20–13 Rough Estimates of CPU Requirements

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, and so on
- 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 10*g* (9.0.4):

SPECint_rate95 baseline = 0.5 * (max # of concurrent operations at peak throughput)

This means 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 10*g* (9.0.4):

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 every 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 these requirements and further assumptions, we developed the following recommendations:

- Disk space: 5 GB to 8 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.

21

Tuning Considerations for the Directory

Once you have completed capacity planning as described in Chapter 20, "Capacity Planning for the Directory", 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

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 for each 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 Performance Tuning Guide* for information about statspack
- *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 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

See Also: "OID Database Statistics Collection Tool (oidstats.sh) Syntax" on page A-133

CPU Usage Tuning

The CPU is perhaps the most important resource available for any software. While Chapter 20, "Capacity Planning for the Directory" 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 four 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 previous table, say we have a 4 CPU box and are able to maintain a peak throughput of 'p' operations every 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 indexes 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 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

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* Database Server

The SGA should be sized based on the available physical memory on the system running Oracle9*i* Database Server.

> **See Also:** *Oracle9i Database Performance Tuning Guide* 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 for each concurrent database connection previously determined.

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 Tuning Guide* in the Oracle Database Documentation Library for general information about balancing and tuning disk I/O

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 for each instance) x
            (# DB Connections for each 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 for each server, for each 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 21-7. 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 sorts on disk.

- 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 Oracle Internet Directory, 10*g* (9.0.4), the directory server entry cache is supported only in the single directory server instance. The benefits of entry caching are maximized when the entry cache hit ratio is very high. Oracle Corporation recommends that the entry cache be used for small-to-medium-sized directory deployments where:

- The working set of directory entries can be completely cached
- The concurrency of clients can be handled by a single directory server instance

Internal benchmarks indicate 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 directory deployments with a larger working set of directory entries and a higher concurrency of clients, it is using the multiprocess directory server instance and the Oracle buffer cache results in greater scalability.

> **See Also:** "Setting System Operational Attributes" on page 5-9 for information about attributes you set to enable and configure entry caching

Optimizing Searches

This section contains these topics:

- Optimizing Searches for Large Group Entries
- Optimizing Searches for Skewed Attributes

Optimizing Searches for Large Group Entries

Searches for group entries with more than a few thousand values for either the `member` or `uniquemember` attribute can have high latency. If you find unacceptably high latency in searches for large group entries with attributes other than `member` and `uniquemember`, then do the following:

1. Set the `orclindexhints` attribute to 1 in the dsaconfig entry. For example, on UNIX, enter the following:

```
ldapmodify -D "cn=orcladmin" -w <passwd> -h ldaphost -p ldapport <<!
dn: cn=dsaconfig,cn=configsets,cn=oracle internet directory
changetype: modify
replace: orclindexhints
orclindexhints: 1
!
```

2. Login to the Oracle Internet Directory database as the ODS user and execute *$ORACLE_HOME*`/ldap/admin/oidbmind.sql`. This creates two bitmap indexes on the `ds_attrstore` table instead of one B-tree index.
3. Restart the Oracle directory server.

Optimizing Searches for Skewed Attributes

To service a typical search request, the directory server sends a SQL statement to the Oracle9*i* Database Server. If a given attribute has very different response times depending on its value, then the attribute is said to be skewed. For example, if searches for `my_attribute=value1` and `my_attribute=value2` have very different response times, then `my_attribute` is said to be a skewed.

You can uniform the response times for searches for such an attribute by adding it as a value of the `orclskewedattribute` attribute, which is in the `dsaconfig` entry. The DN of the `dsaconfig` entry is `cn=dsaconfig,cn=configsets,cn=oracle internet directory`.

By default, the `objectclass` attribute is listed as a value in the `orclskewedattribute` attribute.

Optimizing Searches for Skewed Attributes by Using Oracle Directory Manager

To optimize queries to the database:

1. In the navigator pane, expand Oracle Internet Directory **Servers** and select the directory server instance.
2. In the right pane, select the **Query Optimization** tab. The fields in the **Query Optimization** tab page are listed and described in Table C–36 on page C-34.
3. In the **Query Optimization** tab page, in the **Attributes with Low Cardinality** field, enter the attributes you want to designate as skewed.
4. Choose **Apply**.

Optimizing Searches for Skewed Attributes by Using ldapmodify

To optimize the search for a skewed attribute, you use ldapmodify to add it as a value of the `orclskewedattribute` attribute.

For example, to add `my_attribute` to the `orclskewedattribute` attribute, you would enter the following:

```
ldapmodify -D "cn=orcladmin" -w password -h host -p port <<!
dn: cn=dsaconfig,cn=configsets,cn=oracle internet directory
changetype: modify
add: orclskewedattribute
orclskewedattribute: my_attribute
!
```

Setting the Time Limit Mode

When you set the server operation time limit as described in "Setting System Operational Attributes" on page 5-9, you specified the maximum number of seconds allowed for a search to be completed. To adjust server performance, you can also set the search time limit mode to be either accurate or approximate. If you specify it as accurate, then searches end precisely at the specified number of seconds. If you specify it as approximate, then searches end within a few seconds of the specified number of seconds. In smaller workloads, the latter provides better performance.

Setting the Time Limit Mode by Using Oracle Directory Manager

To set the time limit mode:

1. In the navigator pane, expand Oracle Internet Directory **Servers** and select the directory server instance.
2. In the right pane, select the **Query Optimization** tab.
3. In the **Query Optimization** tab page, in the Time Limit Mode field, select either **Accurate** or **Approximate**.
4. Choose **Apply**.

Setting the Time Limit Mode by Using ldapmodify

To specify the search time limit mode to be either accurate or approximate, you set the `orcltlimitmode` attribute. A value of `0` is accurate, and a value of `1` is approximate. The default value is `1`.

Setting the Timeout for Client/Server Connections

You can specify the amount of idle time allowed for connections between clients and the directory server.

Setting the Timeout for Client/Server Connections by Using Oracle Directory Manager

To set the timeout for client/server connections:

1. In the navigator pane, expand Oracle Internet Directory **Servers** and select the directory server instance.
2. In the right pane, select the **Query Optimization** tab.
3. In the **Query Optimization** tab page, in the **LDAP Connection Timeout** field, enter the maximum number of seconds that the directory client can remain idle before the connection is terminated. The default is `0`, meaning that there is no timeout.
4. Choose **Apply**.

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 (oidstats.sh) Syntax" on page A-133 for instructions on using the OID Database Statistics Collection tool

22

Garbage Collection in Oracle Internet Directory

The term "garbage" refers to any data not needed by the directory but still occupying space on it. This unwanted or obsolete data can eventually fill up the disk and decrease directory performance. The process of removing this unwanted data from the directory is called garbage collection.

This chapter contains these topics:

About the Oracle Internet Directory Garbage Collection Framework

A garbage collector is a background database process that removes unwanted data from the directory. The Oracle Internet Directory garbage collection framework provides a default set of garbage collectors, and enables you to modify them.

This section contains these topics:

- Components of the Oracle Internet Directory Garbage Collection Framework
- How Oracle Internet Directory Garbage Collection Works
- Garbage Collector Entries
- Change Log Purging in Multimaster Replication

Components of the Oracle Internet Directory Garbage Collection Framework

This section describes the components that make up the Oracle Internet Directory garbage collection framework, namely, the garbage collection plug-in and the garbage collectors themselves.

Garbage Collection Plug-in

Garbage collection in Oracle Internet Directory relies on a garbage collection plug-in that receives requests to manage garbage collectors. This plug-in is installed with Oracle Internet Directory, and is enabled by default. The entry for this plug-in is `cn=plugin,cn=subconfigsubentry`.

This plug-in has three triggers:

- The DN of the plug-in trigger used to create a garbage collection job is: `cn=Add PurgeConfig,cn=plugin,cn=subconfigsubentry`.
- The DN of the plug-in trigger used to modify a garbage collection job is: `cn=Modify PurgeConfig,cn=plugin,cn=subconfigsubentry`.
- The DN of the plug-in trigger used to delete a garbage collection job is: `cn=Delete PurgeConfig,cn=plugin,cn=subconfigsubentry`.

> **See Also:** "Oracle Internet Directory Plug-In for Garbage Collection" on page B-15 for a list and descriptions of the attributes of the garbage collection plug-in

Garbage Collectors

Garbage collectors are background database processes that are invoked by the garbage collection plug-in. You can set and manage these behaviors of a garbage collector:

- The subtree in which it purges data
- The time it starts
- The age of the data you want it to purge
- How often it runs
- The type of data you want it to purge
- The number of entries to purge at a time

Predefined Garbage Collectors A default installation of Oracle Internet Directory includes these predefined garbage collectors:

- Audit log garbage collector—Cleans up obsolete entries created for auditing the directory. The container for this garbage collector is `cn=auditlog purgeconfig,cn=purgeconfig,cn=subconfigsubentry`.

  See Also: "Audit Log Garbage Collector" on page B-9

- Change log garbage collector—Cleans up the consumed change log entries in the directory. The container for this garbage collector is `cn=changelog purgeconfig, cn=purgeconfig,cn=subconfigsubentry`.

  See Also: "Change Log Garbage Collector" on page B-10

- General statistics garbage collector—Cleans up obsolete entries created by Oracle Internet Directory Server Manageability for monitoring general statistics of the directory. The container for this garbage collector is `cn=general stats purgeconfig, cn=purgeconfig,cn=subconfigsubentry`.

  See Also: "General Statistics Garbage Collector" on page B-11

- Health statistics garbage collector—Cleans up obsolete entries created by Oracle Internet Directory Server Manageability for monitoring health statistics of the directory. The container for this garbage collector is `cn=health stats purgeconfig, cn=purgeconfig,cn=subconfigsubentry`.

  See Also: "Health Statistics Garbage Collector" on page B-12

- Security and refresh events garbage collector—Cleans up obsolete entries created by Oracle Internet Directory Server Manageability for monitoring security and refresh events of the directory. The container for this garbage collector is `cn=secrefresh events purgeconfig, cn=purgeconfig,cn=subconfigsubentry`.

  See Also: "Security and Refresh Events Garbage Collector" on page B-13

- System resource events garbage collector—Cleans up obsolete entries created by Oracle Internet Directory Server Manageability for monitoring system resource events of the directory. The container for this garbage collector is `cn=sysresource events purgeconfig, cn=purgeconfig,cn=subconfigsubentry`.

  See Also: "System Resource Events Garbage Collector" on page B-14

- Tombstone garbage collector—Cleans up obsolete entries marked as deleted in the directory. The container for this garbage collector is `cn=tombstone purgeconfig, cn=purgeconfig,cn=subconfigsubentry`.

  See Also: "Tombstone Garbage Collector" on page B-15

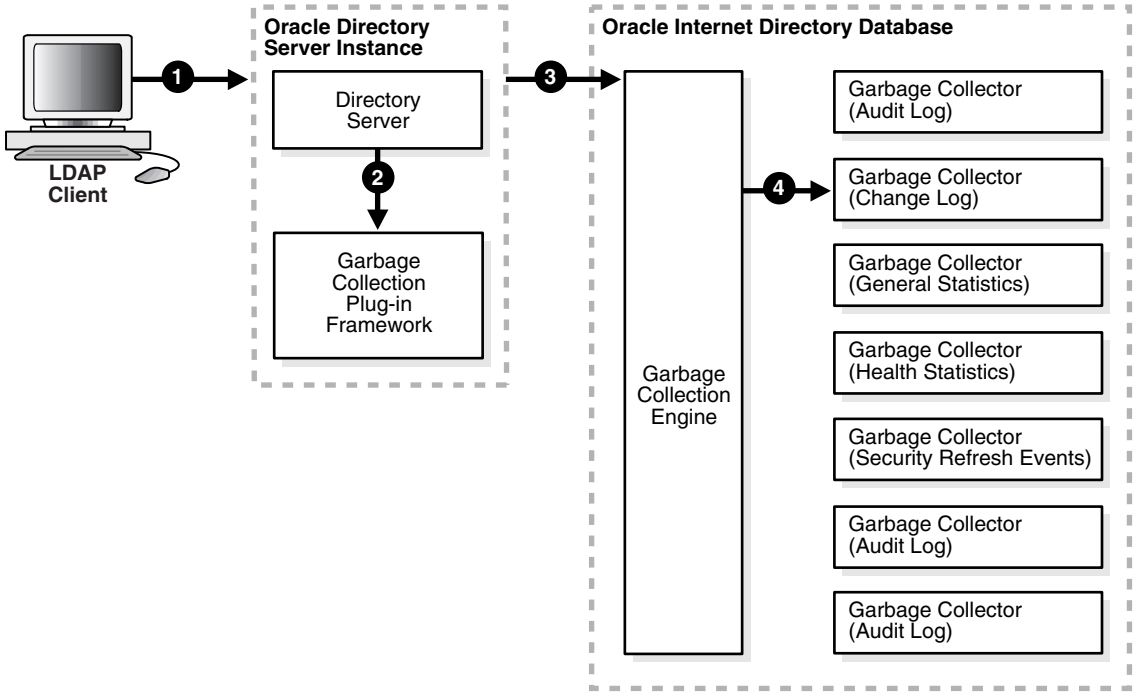
Note: Oracle Corporation recommends that you not delete any of the predefined garbage collectors. Deleting one or more of them can result in the proliferation of obsolete data, eventually exhausting all the available disk space.

You may, however, modify predefined garbage collectors to customize their behavior.

How Oracle Internet Directory Garbage Collection Works

Figure 22–1 shows an example of a garbage collector operation that purges change log entries.

Figure 22–1 Example: Garbage Collection of Change Log Entries



As the example in Figure 22–1 on page 22-5 shows, the garbage collection process is as follows:

1. An LDAP client sends to the directory server a request for a particular garbage collection operation. The operation could be, for example, to purge the entries of tombstones, change logs, or audit logs.
2. The directory server passes the request to the garbage collection plug-in.
3. The garbage collection plug-in sends the request to the garbage collection engine in the Oracle Internet Directory-designated database.
4. The garbage collection engine triggers the corresponding garbage collector—in this case, the change log garbage collector. The garbage collector runs as a background database process according to the parameters specified in its configuration set entry.

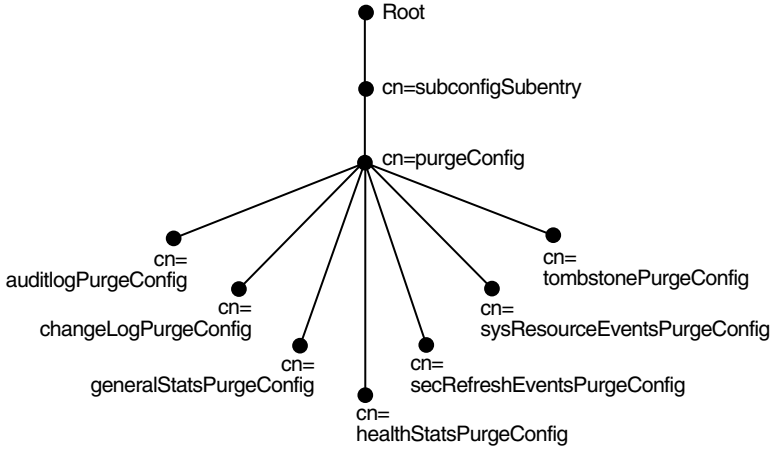
Garbage Collector Entries

Garbage collector entries, each with attributes specifying how it is to behave, are located in the entry `cn=purgeconfig`, which is located immediately below the entry `cn=subconfigsubentry`.

> **See Also:** Table B–8, " Garbage Collection Configuration Parameters" on page B-8 for a description of each garbage collector attribute

Figure 22–2 shows the location of these entries.

Figure 22–2 Garbage Collection Entries in the DIT



Change Log Purging in Multimaster Replication

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.

  Note that this method always respects the rules of change log purging. If, at the specified time, a change has not yet been applied, this method does not purge it.

See Also:

- "Viewing and Modifying Directory Replication Server Configuration Parameters" on page 25-36
- "Modifying Directory Replication Server Configuration Parameters by Using Command-Line Tools" on page 25-37

Modifying Oracle Internet Directory Garbage Collectors

This section contains these topics:

- Modifying a Garbage Collector by Using Oracle Directory Manager
- Modifying a Garbage Collector by Using Command-Line Tools

Modifying a Garbage Collector by Using Oracle Directory Manager

To modify a garbage collector:

1. In the navigator pane, expand in succession Oracle Internet Directory **Servers**, ***directory server instance***, **Garbage Collection Management**, then select the garbage collector you want to configure. The Garbage Collector Window appears in the right pane.
2. In the **Garbage Collector** window, enter the values for this garbage collector. These fields are described in Table C–7 on page C-5.
3. Choose **Apply**.

Modifying a Garbage Collector by Using Command-Line Tools

This section provide examples of how to modify garbage collectors by using command-line tools. The garbage collection attributes that you can modify are listed in "Schema Elements for Predefined Garbage Collectors" on page B-8.

Example 1: Modifying a Garbage Collector

Suppose that you want the Tombstone garbage collector to run immediately. The LDIF would look like this:

```
dn: cn=tombstone purgeconfig, cn=purge config, cn=subconfigsubentry
changetype:modify
replace: orclpurgerun
orclpurgerun: 1
```

Load this entry with ldapmodify.

```
ldapmodify -h hostname -p port# -D username -w passwd -f  file_name_of_defined_
entry
```

Example 2: Disabling a Garbage Collector Change Log

Suppose that you want to disable changelog Garbage Collector.

```
dn: cn=changelog purgeconfig, cn=purgeconfig, cn=subconfigsubentry
changetype: modify
replace: orclpurgeenable
orclpurgeenable: 0
```

Load this entry with ldapmodify.

```
ldapmodify -h hostname -p port# -D username -w passwd -f  file_name_of_defined_
entry
```

Enabling and Disabling Logging for Oracle Internet Directory Garbage Collectors

This section contains these topics:

- Enabling Logging for Oracle Internet Directory Garbage Collectors
- Disabling Logging for Oracle Internet Directory Garbage Collectors

Enabling Logging for Oracle Internet Directory Garbage Collectors

If you enable logging for garbage collectors, then the directory server writes the information into a file in the file system. This information includes:

- The job identifier
- A job description of the garbage collector
- The number of entries purged

To enable logging of garbage collection information:

1. Set the `orclpurgedebug` attribute to 1.
2. Set the `orclpurgefilename` attribute to a valid file name for the log file
3. Set the `orclpurgefileloc` attribute to the path name of the directory in which the log file is located.

4. Enable PL/SQL I/O. To do this:

   a. In the database initialization file, include the following:

      ```
      UTL_FILE_DIR=PATH_NAME
      ```

      where *PATH_NAME* is the one you specified in Step 3.

   b. Restart the database.

      See Also: The section on the UTL_FILE_DIR parameter type in the *Oracle9i Database Reference*

Disabling Logging for Oracle Internet Directory Garbage Collectors

To disable logging of garbage collection information, set the `orclpurgedebug` attribute to `0`.

23

Migration of Data from Other Directories

This chapter explains how to migrate data from both LDAP Version 3-compatible directories and application-specific directories into Oracle Internet Directory. It also explains how to migrate an existing directory into the default directory structure explained in Chapter 19, "Deployment of Oracle Identity Management Realms".

This appendix contains these topics:

- Migrating Data from LDAP-Compliant Directories
- Migrating User Data from Application-Specific Repositories
- Migrating an Existing Directory into the Default Directory Structure

Migrating Data from LDAP-Compliant Directories

This section 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:** RFC 2849 of the IETF, available for download at: `http://www.ietf.org`

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 23-2

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 "The schemasync Tool Syntax" on page A-125.

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 10*g* (9.0.4) 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 manually re-enter or upload hashed `userPassword` information separately into the directory. Be sure that the passwords comply with the Oracle Internet Directory password policies and are in clear text.

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 1.3.2.2-1 or later. Visit:
  `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit:
  `http://www.datafocus.com/`

See Also: "bulkload Syntax" on page A-45 for instructions on how to use the bulkload check mode

Migrating User Data 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

The Intermediate Template File

To enable this migration to happen, the Oracle Directory Provisioning Integration Service requires the application-specific repository to export its data to an intermediate template file. Records in this template file are not in pure LDIF; they contain substitution variables that have to do with, for example, the location in the directory where the information is finally to reside. The application leaves these variables undefined, so that you, the directory administrator can define them later on.

To convert the user data from this intermediate template file into proper LDIF, you use the OID Migration Tool (ldifmigrator). 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. Exporting the application-specific data as an intermediate template file

2. You, the directory administrator, using the OID Migration Tool (ldifmigrator) to read these partial LDIF entries and convert them to pure LDIF entries based on the deployment choices
3. You, the directory administrator, loading the data, now in pure LDIF, into Oracle Internet Directory
4. The application completing the migration process according to its own specifications

Reconciling Data in Application Repository with Data Already in Oracle Internet Directory

The data you are migrating from an application-specific repository may already reside in Oracle Internet Directory. If this is the case, then you can reconcile differences between the two directories by using the reconciliation feature of the OID Migration Tool (ldifmigrator).

See Also:

- "Load Capability" on page A-140
- "Reconcile Capability" on page A-140 for information about the reconciliation feature of the OID Migration Tool

Tasks For Migrating Data from Application-Specific Repositories

To migrate data from application-specific repositories, you create an intermediate template file, then run the OID Migration Tool.

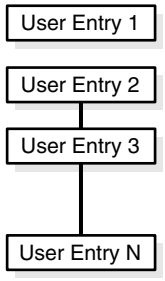
Task 1: Create an Intermediate Template File

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`.

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 Migration Tool (ldifmigrator) assigns all of these users to the default identity management realm, which corresponds to the enterprise itself.

Figure 23–1 shows the overall structure of the intermediate template file containing user entries.

Figure 23–1 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 *$ORACLE_HOME*/ldap/schema/oid.

Attributes in User Entries Each user entry has mandatory and optional attributes.

Table 23–1 lists and describes the mandatory attributes in a user entry.

Table 23–1 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 contain the cn attribute.
sn	Surname—that is, the last name—of the user
objectclass	Object classes the entry should minimally belong to: inetOrgPerson and orclUserV2

See Also:

- IETF Request for Comments 2798: "Definition of the `inetOrgPerson` LDAP Object Class," available at `http://www.ietf.org`, for a description of each attribute in the `inetOrgPerson` object class
- "Optional Attributes of the orclUserV2 Object Class" on page B-17

Task 2: Run the OID Migration Tool

Once you have set up the intermediate template file, the OID Migration Tool 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 the Oracle Directory Synchronization Service.

See Also: "The OID Migration Tool (ldifmigrator) Syntax" on page A-135 for instructions about using the OID Migration Tool

Migrating an Existing Directory into the Default Directory Structure

During an Oracle Internet Directory installation, Oracle Universal Installer creates a default schema and directory information tree (DIT). This default DIT framework, described in Chapter 19, "Deployment of Oracle Identity Management Realms", is flexible: you can modify it to suit the needs of your deployment.

If you have a directory with an already-established structure, and you want to migrate the data from that directory into the default directory structure environment, then follow the instructions in this section. This section contains these topics:

- The Default Directory Structure
- Changing the Location of Users or Groups in the Oracle Context of the Default Identity Management Realm

The Default Directory Structure

In Oracle Internet Directory 10*g* (9.0.4), the following directory elements are created by default:

- Root Oracle Context (`cn=OracleContext`)—This is the container where Oracle products store enterprise-wide configuration data.
- Default identity management realm (`dc=`*`dns_domain_of_host`*`,dc=com`)—This is the container under which Oracle products expect to find enterprise users and groups. It approximates the enterprise DIT structure. For example, if Oracle Internet Directory is installed on a computer whose host name is: `my_computer.us.my_company.com`, then the default identity management realm created at installation of Oracle Internet Directory would be dc=us,`dc=my_company,dc=com`. Oracle products expect to find all users under the container `cn=users,dc=us,dc=my_company,dc=com` and all groups under `cn=groups,dc=us,dc=my_company,dc=com`. In addition to creating the default identity management realm entry, the Oracle Internet Directory Configuration Assistant stores a pointer to it in the Root Oracle Context so that other Oracle Internet Directory-enabled components can bootstrap themselves.

You can change this default identity management realm to suit your deployment requirements—for example, to store all of enterprise users in a different container.

Changing the Location of Users or Groups in the Oracle Context of the Default Identity Management Realm

To change the location of users or groups in your Oracle Context, you would make the appropriate pointer change in your default identity management realm entry. The following example of an LDIF file changes the location of users or groups in the Oracle Context to `o=my_company,dc=com`:

```
dn: cn=common,cn=products,cn=oracleContext,dc=default_subscriber_name,dc=com
changetype: modify
replace: orclCommonUserSearchBase
orclCommonUserSearchBase: o=my_company,dc=com
```

Part V

Directory Replication and High Availability

This part provides detailed discussions of replication and high availability and how to plan and manage them. It contains these chapters:

24

Directory Replication Concepts

In "Directory Replication" on page 2-23, you saw an overview of replication. This chapter provides a closer look. It contains these topics:

- About Directory Replication
- Full and Partial Directory Replication
- Directory Replication Groups
- Included and Excluded Naming Contexts
- Replication Agreements
- Replication Configuration Objects in the Directory
- Replication Security
- Change Logs in Directory Replication
- Multimaster Replication
- Fan-Out and Partial Replication
- Rules for Partial Replication Filtering

> **See Also:** Chapter 25, "Oracle Directory Replication Administration" for information on managing replication

About Directory Replication

This section briefly introduces some of the basic concepts of replication. The other sections in this chapter explain these concepts in further detail.

Replication is the process of copying and maintaining the same naming contexts on multiple directory servers. It improves performance by providing more servers to handle queries, and reliability by eliminating risks associated with a single point of failure.

Replication can be either full or partial.

Full replication involves propagating the entire DIT to another node.

Partial replication involves propagating one or more subtrees, rather than the entire DIT, to another node.

The directory servers that participate in the replication of a given naming context form what is called a directory replication group (DRG). The relationship among the directory servers in a DRG is represented on each node by a special directory entry called a replication agreement.

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

A directory replication group can be either single-master, multimaster, or fan-out.

A single-master replication group has only one supplier replicating changes to one or more consumers. Only the supplier can be updated, and consumers are read-only.

Multimaster replication, also called peer-to-peer or *n*-way replication, enables multiple sites, acting as equals, to manage groups of replicated data. In a multimaster replication environment, each node is both a supplier and a consumer node, and the entire directory is replicated on each node.

A fan-out replication group, also called a point-to-point replication group, has a supplier replicating directly to a consumer. That consumer can then replicate to one or more other consumers. The replication can be either full or partial.

In a directory replication group, the protocol for transferring data between nodes can be based on either Oracle9*i* Advanced Replication or LDAP.

Full and Partial Directory Replication

This section contains these topics:

Full Directory Replication

Full replication involves propagating the entire DIT to another node. This type of replication ensures the high availability of the entire directory. You can also use it to distribute operations on the entire directory among different nodes.

Full replication can be based on either Oracle9*i* Advanced Replication or LDAP.

Partial Directory Replication

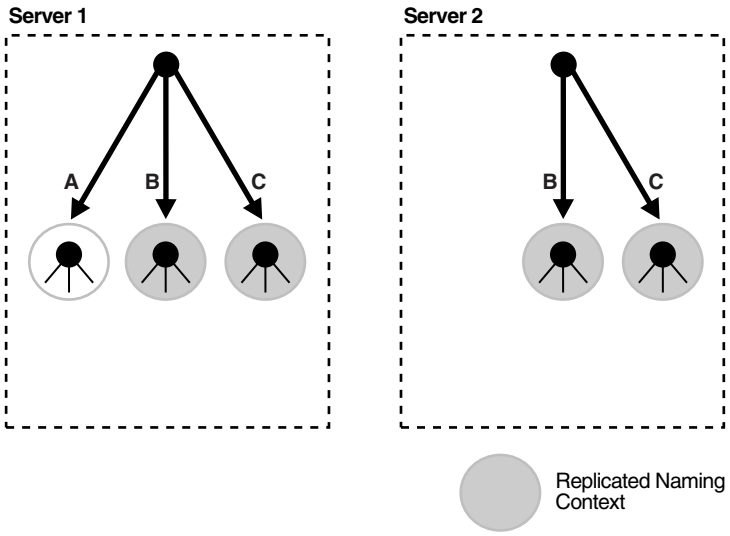
Partial replication enables you to propagate one or more subtrees, rather than the entire DIT, to another node. Decentralizing a directory in this way enables you to balance the workload between servers and build a highly available distributed directory complete with fault tolerance and failover. Because it brings data closer to the client, partial replication reduces response time and improves performance. You can configure partial replication by using the Replication Environment Management Tool.

Partial replication is LDAP-based only. It does not use Oracle9*i* Advanced Replication.

See Also: "The Replication Environment Management Tool" on page A-62

Figure 24–1 shows an example of partial replication.

Figure 24–1 Example of Partial Replication



In a partial replication scenario, one or more naming contexts, but not the entire directory, are replicated. For example, in Figure 24–1, Server 1 contains three naming contexts: A, B, and C. Naming contexts B and C are replicated to Server 2, but naming context A is not.

Table 24–1 compares the two types of replication.

Table 24–1 Comparison of Full and Partial Replication

Full Replication	Partial Replication
Propagates an entire directory to other nodes	Propagates just part of the directory—for example, one or more, but not all, naming contexts—to other nodes
Propagates to a limited number of nodes	Propagates to an unlimited number of nodes
In a multimaster environment, allows a consumer to receive changes from more than one supplier	In a single-master environment, allows a consumer to receive changes from only one supplier

Directory Replication Groups

This section contains these topics:

- Data Transfer Between Nodes in a Directory Replication Group
- Single-Master Replication Groups
- Multimaster Replication Groups
- Fan-Out Replication Groups
- Types of Directory Replication Compared
- Multimaster Replication with Fan-Out

See Also: "Replication Agreements" on page 24-12

Data Transfer Between Nodes in a Directory Replication Group

In a directory replication group, the protocol for transferring data can be based on either Oracle9*i* Advanced Replication or LDAP. Table 24–2 shows how each type handles various features of replication and points to sections of this chapter with more details.

Table 24–2 Types of Data Transfer Between Nodes in a Directory Replication Group

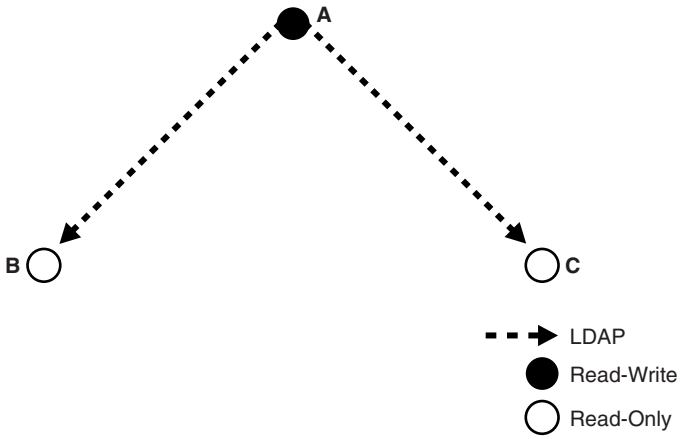
Feature	LDAP-Based Replication	Oracle9*i* Advanced Replication-Based Replication	More Information
Change propagation	Change propagation from supplier to consumer happens over LDAP	Change propagation from supplier to consumer happens by using Oracle9*i* Advanced Replication	"Change Logs in Directory Replication" on page 24-19
Replica type supported	Read-only full replica Read-only partial replica	Read/write full replica	"Full Directory Replication" on page 24-3 "Partial Directory Replication" on page 24-3
Configuration supported	Single-master replication Fan-out replication	Multimaster replication Single-master replication, by switching all masters in a multimaster configuration except one to read-only mode.	"Single-Master Replication Groups" on page 24-6 "Multimaster Replication Groups" on page 24-6 "Fan-Out Replication Groups" on page 24-7

Single-Master Replication Groups

A single-master replication group has only one supplier replica supplying changes to one or more consumers. Clients can update only the master replica, and can only read data on any of the consumers.

Figure 24–2 shows a single-master replication environment.

Figure 24–2 Example of Single-Master Replication



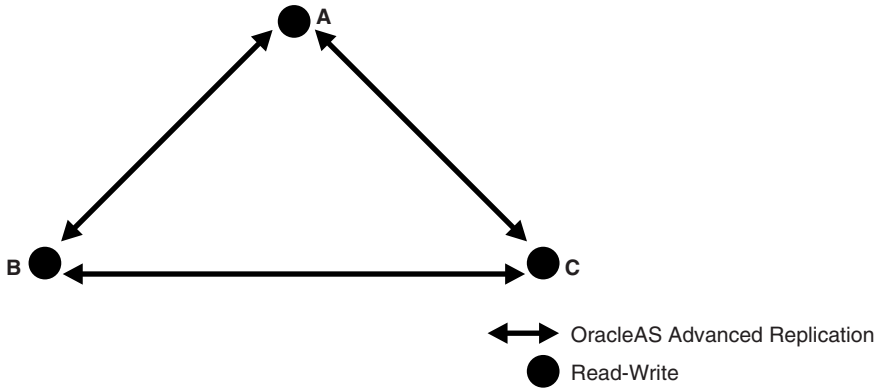
In Figure 24–2, each bullet represents a node of Oracle Internet Directory. Node A is a supplier that replicates consumer nodes B and C. Node A is read/write, and Nodes B and C are read-only. The data transfer protocol is LDAP.

Multimaster Replication Groups

A multimaster replication group, also called a peer-to-peer or *n*-way replication group, has two or more nodes acting as equals to manage groups of replicated data. In a multimaster replication group, each directory server is both a supplier and a consumer of changes, and the entire directory is replicated on each node.

The example in Figure 24–3 shows three nodes—A, B, and C—that update each other in a multimaster replication group.

Figure 24–3 Example of Multimaster Replication



In Figure 24–3, each node is read/write, and the data transfer protocol is based on Oracle9*i* Advanced Replication.

Note: Multimaster replication is the only replication mechanism supported in Oracle Application Server Single Sign-On as described in the section "Configuring Oracle Application Server Single Sign-On for Replication" in the chapter on high availability in the *Oracle Application Server Single Sign-On Administrator's Guide*

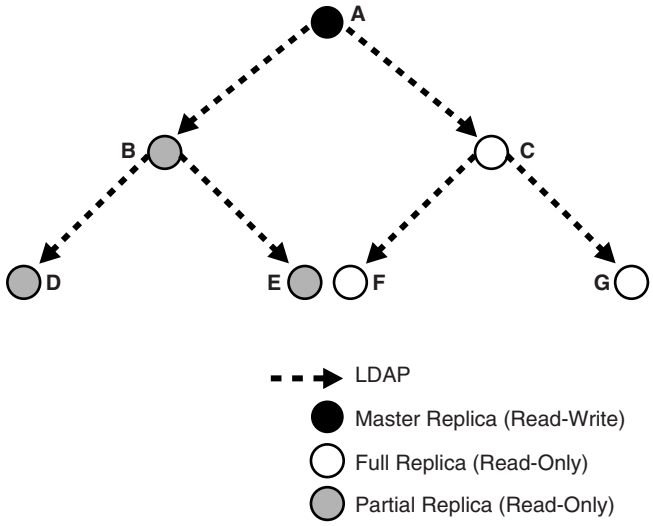
See Also: "Multimaster Replication" on page 24-20 for more information about multimaster replication

Fan-Out Replication Groups

A fan-out replication group, also called a point-to-point replication group, has a supplier replicating directly to a consumer. That consumer can then supply the same data to one or more other consumers. The replication can be either full or partial.

Figure 24–4 shows a fan-out replication environment.

Figure 24–4 Example of Fan-Out Replication



In Figure 24–4, supplier A replicates to two consumers, B and C. Consumer node B contains a partial replica of A, whereas consumer node C contains a full replica of A. Both consumer nodes B and C are read-only.

Each of these nodes, in turn, serves as a supplier that replicates data to two other consumers: Node B partially replicates to nodes D and E, and node C fully replicates to nodes F and G.

In fan-out replication, nodes transfer data by using LDAP.

Types of Directory Replication Compared

Table 24–3 compares multimaster, single-master, and fan-out replication.

Table 24–3 Multimaster. Single-Master, and Fan-Out Replication Compared

Multimaster Replication	Single-Master Replication	Fan-out Replication
Uses only Oracle9*i* Advanced Replication	Uses LDAP-based replication	Uses LDAP-based replication
Updates can be made on any node, whether supplier or consumer.	Updates can be made on the supplier only. Changes to a consumer can be propagated to other fan-out consumers, but not back to the supplier.	Updates can be made on the supplier only. Changes to a consumer can be propagated to other fan-out consumers, but not back to the supplier. This is also true when the LDAP-based replica node is read/write.

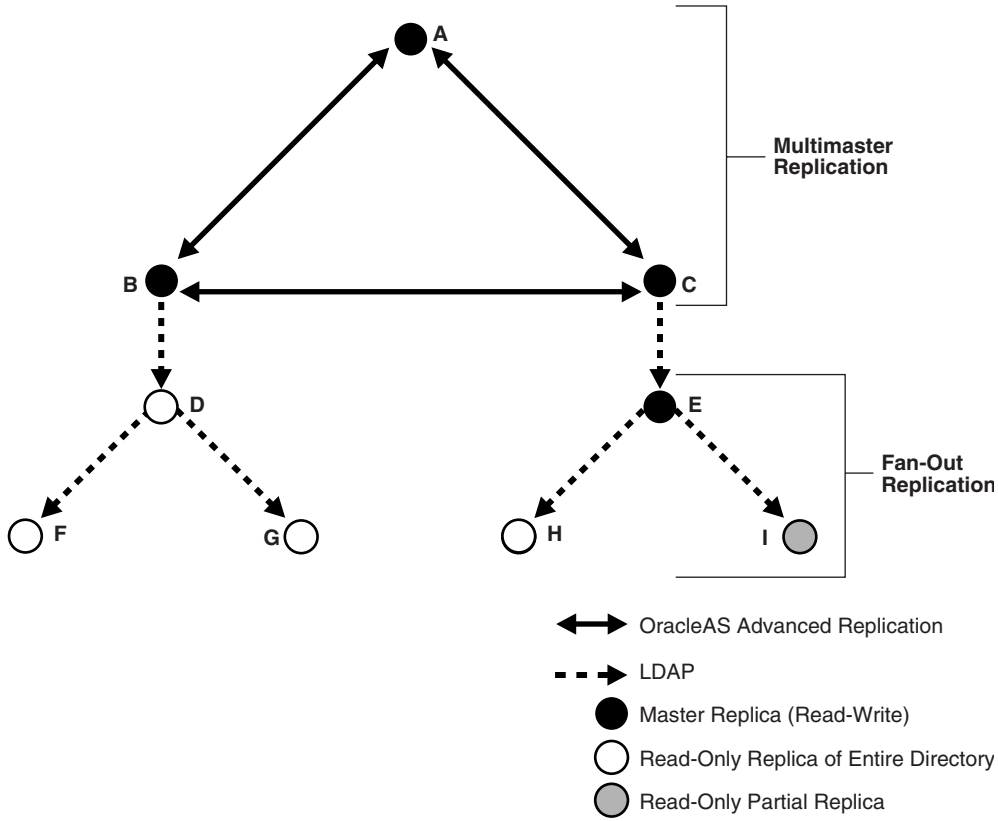
Multimaster Replication with Fan-Out

Oracle Internet Directory Release 9.0.4 enables any node in a multimaster replication group to supply all or part of its data to a read-only consumer. This consumer can, in turn, supply data to other consumers in a fan-out configuration. Within the multimaster replication agreement, data transfer between the nodes occurs by way of Oracle9*i* Advanced Replication. Within the fan-out replication agreement, data transfer from supplier to consumer occurs by way of LDAP.

Note: If an LDAP-based replica is read/write, then changes on this node propagate to consumers, but not to suppliers.

Figure 24–5 shows an example of multimaster replication used in conjunction with fan-out replication.

Figure 24–5 Example of Multimaster Replication with Fan-Out



In the example in Figure 24–5, nodes A, B, and C form a multimaster replication group. They transfer data between them by using Oracle9*i* Advanced Replication.

Node B supplies changes to Node D, a read-only replica of the entire directory. Node D, in turn, supplies changes to Nodes F and G by using LDAP-based replication. Both Nodes F and G are read-only replicas of the entire directory. Similarly, Node C supplies changes to node E, a read/write replica of the entire directory. Node E, in turn supplies changes to Node H, a read-only replica of the entire directory, and Node I, a read-only partial replica, by using LDAP-based replication.

See Also: "Fan-Out and Partial Replication" on page 24-33 for more information about fan-out replication

Included and Excluded Naming Contexts

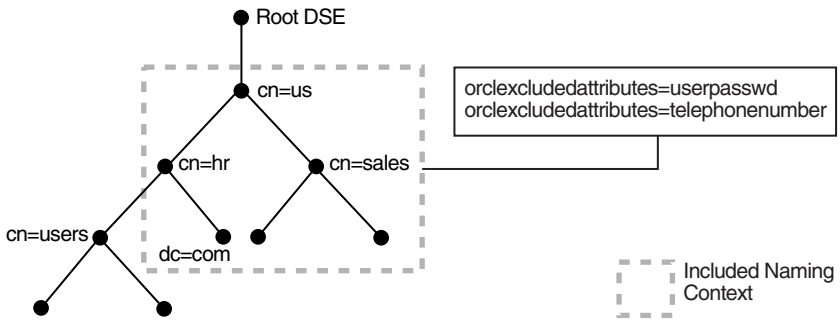
You can specify a given naming context for replication, but exclude from that replication one or more of the subtrees within that naming context. You can also exclude from replication one or more of the attributes in that naming context.

In LDAP-based replication, naming contexts are excluded from replication by default.

In replication based on Oracle9*i* Advanced Replication, naming contexts are included by default. To exclude a naming context from replication, specify it in the `orclexcludednamingcontext` attribute in the replication agreement `orclagreementid=000001`.

Figure 24–6 on page 24-11 and the accompanying text further exemplify the use of the naming context container and its objects.

Figure 24–6 Example of a Naming Context Container and Its Objects



In Figure 24–6, the naming context included for replication is `cn=us`. Within that naming context, one subtree, namely `cn=users`, `cn=hr`, `cn=us` is excluded from replication. Moreover, two of the attributes of the `cn=us` naming context are excluded from replication—namely, `userpassword` and `telephonenumber`.

See Also: The Replication Naming Context Container Entry on page 24-14

Replication Agreements

A replication agreement is a special entry containing information about the relationship among servers in a DRG. In Oracle Internet Directory, all such entries reside under the container entry `cn=replication configuration` located at the root DSE. This entry resides on each node in a DRG, and contains all replication information for that node.

There are two kinds of replication agreements: those for multimaster replication groups, and those for single-master replication groups.

This section contains these topics:

- Multimaster Replication Agreements
- Single-Master Replication Agreements
- Examples of Replication Configuration Objects in the Directory

Multimaster Replication Agreements

For a multimaster replication group, replication agreements are based on Oracle9*i* Advanced Replication. The replication agreement on each node lists all of the nodes in the group. It is identical on each node except for local options such as partitioned naming contexts on the local directory server.

The entry for this kind of replication agreement resides immediately below the `cn=replication configuration` container entry. For example, the DN of such an agreement can look like this: `orclagreementID=000001,cn=replication configuration`.

Single-Master Replication Agreements

Unlike replication agreements for multimaster replication groups, replication agreements for single-master replication groups are LDAP-based. For each fan-out replication group there is one replication agreement for each supplier-consumer relationship.

The entry for this kind of replication agreement resides immediately below the node that serves as the supplier. Thus, the DN of the replication agreement as found on a supplier node is:

```
orclagreementID=unique_identifier_of_the_replication_
agreeement,
orclReplicaID=unique_identifier_of_supplier_node,
cn=replication configuration
```

Similarly, the DN of the replication agreement as found on a consumer node is:

```
orclagreementID=unique_identifier_of_the_replication_
agreeement,
orclReplicaID=unique_identifier_of_supplier_node,
cn=replication configuration
```

In a fan-out replication agreement, you can tell which node the agreement entry is associated with by looking at its parent. For example, look at the following replication agreement entry.

```
orclagreementID=000002,orclReplicaID=node_A,cn=replication
configuration
```

In this example, you can determine that the replication agreement represented by `orclagreementID=000002` is associated with node A. This is because the parent of `orclagreementID=000002` is `orclReplicaID=node_A`.

Note: The container entry `cn=replication configuration` is replicated on all nodes, but may not be identical on all nodes.

See Also: "The Replication Naming Context Container Entry" on page 24-14

Replication Configuration Objects in the Directory

This section describes the objects in the directory that contain replication configuration information. It contains these topics:

The Replication Configuration Container

All replication information for a node resides in the container `cn=replication configuration` located at the root DSE. This entry resides on each node in a DRG.

The Replica Subentry

This subentry is created at installation under the replication configuration container. It contains attributes that identify and define the characteristics of the node it represents.

This subentry is associated with the object class `orclReplicaSubentry`. It contains the attribute `orclreplicaID` whose value specifies the name of the replica subentry. It is unique to each directory node, and matches that of the `orclreplicaID` attribute at the root DSE. For example, in Figure 24–9 on page 24-17, a replica subentry is represented by `orclReplicaID=`*`UID_of_node_D`*`,cn=replication configuration`.

> **See Also:** Table B–31 on page B-38 for descriptions of the attributes of the replica subentry.

The Replication Agreement Entry

This entry contains attributes that define the replication agreement between a consumer and a supplier. It resides under the replication configuration entry, and is associated with the `orclReplAgreementEntry` object class. The naming attribute of this entry is `orclagreementID`. For example, in Figure 24–9 on page 24-17, a replication agreement entry is represented by `orclagreementID=`000003`,orclReplicaID=`*`UID_of_node_D`*`,cn=replication configuration`.

> **See Also:** Table B–32 on page B-38 for descriptions of the attributes of the replication agreement entry

The Replication Naming Context Container Entry

This entry contains all the LDAP naming context objects. These objects specify what is to be either included in or excluded from replication to an LDAP-based partial replica.

This entry has the RDN `cn=replication namecontext`, and it is created below the `orclagreementID` entry at installation.

A replication naming context contains these objects:

- `orclincludednamingcontexts`—The root of the naming context to be replicated
- `orclexcludednamingcontexts`—Within the included naming context, the root of a subtree to be excluded from replication

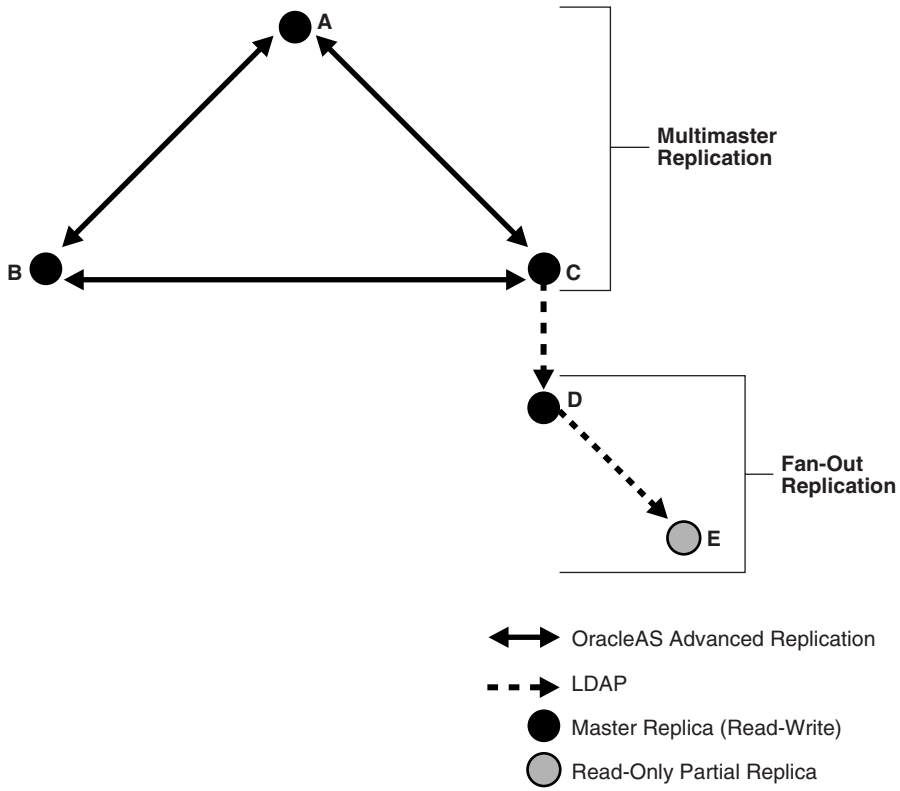
- `orclexcludedattributes`—Within the included naming context, an attribute to be excluded from replication

  See Also: Table B–33 on page B-40 for descriptions of the attributes in the replication naming context container entry

Examples of Replication Configuration Objects in the Directory

The examples of replication objects in this section rely on the replication environment shown in Figure 24–7.

Figure 24–7 Example: Multimaster Replication and Fan-Out Replication



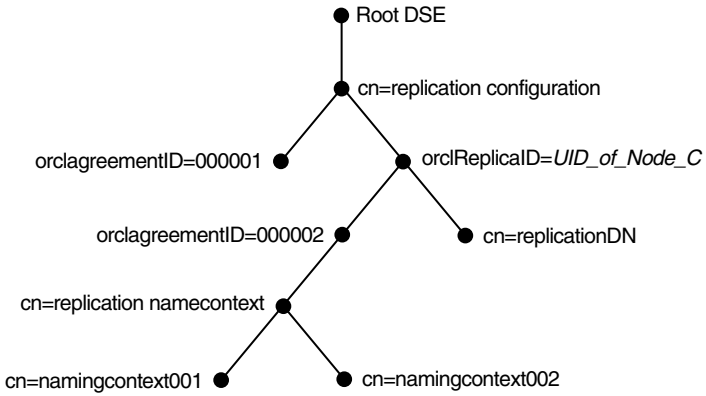
In Figure 24–7, nodes A, B, and C form a multimaster replication group. Node C replicates to a fourth node, D, which, in turn, fans out to Node E.

The replication agreements in this environment are as follows:

- Node A has one replication agreement representing its multimaster relationship with nodes B and C.
- Node B has one replication agreement representing its multimaster relationship with nodes A and C.
- Node C has two replication agreements, the first representing its multimaster relationship with nodes A and B, the second representing its relationship to node D in which it serves as the supplier and node D is the consumer.
- Node D has two replication agreements, one representing its relationship to the supplier node C, from which it consumes changes, the other representing its relationship to consumer node E for which it is the supplier.

Figure 24–8 shows the replication objects in the DIT that pertain to node C in Figure 24–7 on page 24-15.

Figure 24–8 Example: Replication Configuration Entries for Node C



For node C, the entry `cn=replication configuration` at the root DSE contains these RDNs:

- `orclagreementID=000001`: The multimaster replication agreement in which node C participates with nodes A and B.
- `orclReplicaID=`*`UID_of_node_C`*: Unique identifier of node C that contains information about it.
- `orclagreementID=000002`: Unique identifier of the relationship between supplier node C and consumer node D. You know that, in this case,

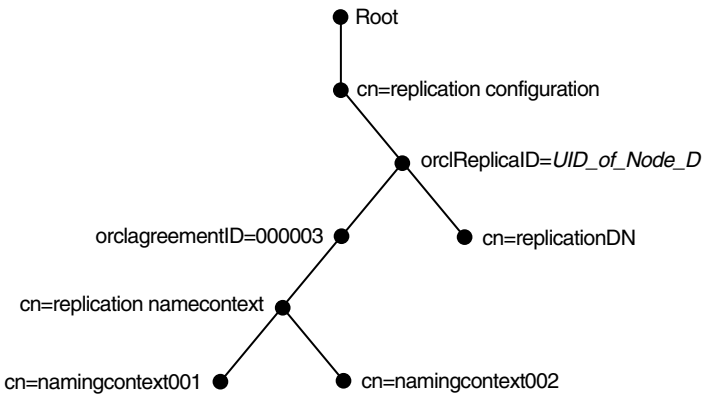
`orclagreementID=000002` is the replication agreement of the supplier node C because node C is its parent.

This entry contains the attribute `orclreplicaDN`, the value of which is the DN of consumer node D with which node C has the replication agreement.

- `cn=replicationDN`: The bind DN that the directory replication server on node C uses to bind to the directory server.
- `cn=replication namecontext`: Container of information about naming contexts that are included in replication.
- `cn=namingcontext001` and `cn=namingcontext002`: The actual objects that are included in or excluded from replication. In the naming context included for replication, you can specify one or more subtrees to be excluded from replication. In that same included naming context, you can specify particular attributes to be excluded from replication.

Figure 24–9 shows the replication agreement entry in the DIT that pertains to node D in Figure 24–7 on page 24-15.

Figure 24–9 Example: Replication Configuration Entries for Node D



For node D, the entry `cn=replication configuration` at the root DSE contains these RDNs:

- `orclReplicaID=`*`UID_of_node_D`*: Unique identifier of node D and contains information about it.
- `orclagreementID=000003`: Unique identifier of the relationship between supplier node D and consumer node E. You know that, in this case,

`orclagreementID=000003` is the replication agreement of the supplier node D because node D is its parent.

This entry contains the attribute `orclreplicaDN`, the value of which is the DN of consumer node E with which node D has the replication agreement.

- `cn=replicationDN`: Bind DN that the directory replication server on node D uses to bind to the directory server.
- `cn=replication namecontext`: Container of information about naming contexts that are included in replication.
- `cn=namingcontext001` and `cn=namingcontext002`: Objects specifying naming contexts to be included in replication. In the naming context included in replication, you can specify one or more subtrees or particular attributes to be excluded from replication.

Replication Security

This section contains these topics:

- Authentication and the Directory Replication Server
- Secure Sockets Layer (SSL) and Oracle Internet Directory Replication

Authentication and the Directory Replication Server

Authentication is the process by which the Oracle directory replication server establishes the true identity of itself when connecting to the directory server. It occurs when an LDAP session is established by means of an ldapbind operation.

It is important that the directory replication server be properly authenticated before it is allowed access to the directory.

The directory replication server uses a unique identity and a password to authenticate with the directory server. The identity of the directory replication server is of the form `cn=replication dn,orclreplicaid=`*`unique_identifier_of_node`*`,cn=replication configuration`.

When it starts, the directory replication server reads its identity and password from an Oracle Internet Directory secure wallet, and uses these credentials for authentication. If you want to change the password for the replication bind DN, then you must use the Replication Environment Management Tool `-pchgpwd` option.

See Also: "The Replication Environment Management Tool" on page A-62

Secure Sockets Layer (SSL) and Oracle Internet Directory Replication

You can deploy Oracle Internet Directory replication with or without SSL.

To configure LDAP-based replication to use SSL encryption, in the `orclReplicaURI` attribute, which contains the supplier contact information, specify the port number of the SSL port.

To configure Oracle9*i* Advanced Replication to use SSL encryption, use Oracle Advanced Security.

See Also: *Oracle Advanced Security Administrator's Guide*for information on how to configure Oracle9*i* Advanced Replication to use SSL encryption

Change Logs in Directory Replication

Oracle Internet Directory records each change as an entry in the change log store. The directory replication server of the consumer retrieves changes residing in the change log store of the supplier and applies them to the consumer.

Each entry in the change log store—that is, each change log object—has a unique change number. The consumer keeps track of the change number of the last change it applied, and it retrieves from the supplier only those changes with numbers greater than that of the last change it applied.

- In an LDAP-based replication agreement, the directory replication server stores the last change number it applied in the `orlclastappliedchangenumber` attribute of the replication agreement entry.
- In a replication agreement based on Oracle9*i*Advanced Replication, the directory replication server stores the last change number it applied in the `changenumber` attribute of the `changestatus` entry. This entry looks like this: `changenumber=`*`last_applied_change_number`*`, supplier=`*`supplier_node`*`, consumer=`*`consumer_node`*. For example, if the last change a consumer applied had a number of 250, then subsequent changes it retrieves from that supplier would need to have numbers greater than 250.

Multimaster Replication

This section gives a detailed look at multimaster replication. A multimaster directory replication group has multiple nodes acting as equals to manage groups of replicated data. This section contains these topics:

- Oracle9i Advanced Replication
- Architecture for Multimaster Replication
- Conflict Resolution in Multimaster Replication
- The Multimaster Replication Process

> **See Also:** "Managing Replication" on page 25-35 for information about how to configure replication agreements

Oracle9*i* Advanced Replication

In Oracle Internet Directory replication, the transport of update information between nodes in a replication agreement is managed by Oracle9*i* Advanced Replication, a store-and-forward transport feature available in Oracle9*i*. Advanced Replication enables you to synchronize database tables across two Oracle databases.

Oracle9*i* Advanced Replication:

- Stores local changes and periodically propagates them in batches to consumers. The consumer replication servers apply the remote changes to their own local directory servers, and then purge the applied remote changes from their local stores.
- Enables read and update access to directory tables anywhere in the Oracle9*i* replication group. Typical Advanced Replication configurations use row-level replication.
- Provides proven network tolerance. Data transfer can be controlled and monitored by Oracle Enterprise Manager Application Server Control. Such manageability allows a high degree of flexibility in how the data transfer is scheduled.

Note: The Oracle Application Server Single Sign-On database schema that resides in the same database as Oracle Internet Directory is also replicated by using Oracle9*i* Advanced Replication.

See Also:

- The section "Configuring Oracle Application Server Single Sign-On for Replication" in the chapter on high availability in the *Oracle Application Server Single Sign-On Administrator's Guide*
- *Oracle9i Advanced Replication* in the Oracle Database Documentation Library for information about Oracle9*i* Advanced Replication

Architecture for Multimaster Replication

Typical Advanced Replication configurations use asynchronous data propagation—that is, suppliers write their changes to change logs, and then regularly send batched changes to other consumers. Consumers receive the change log data, then reproduce the changes locally.

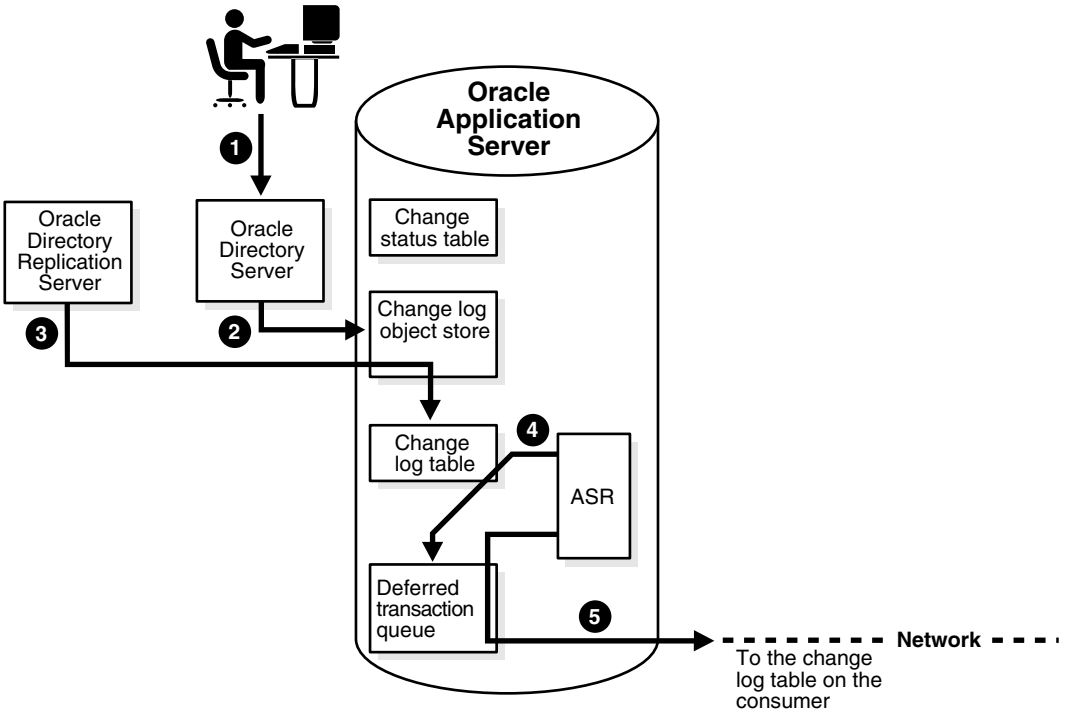
When you configure replication, you specify which nodes in a replication group share changes. Regardless of the number of nodes you introduce into the replication environment, the basic architecture for replication remains the same. Local changes are distributed to a **remote master site (RMS)** where 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 Multimaster Replication Process on the Supplier Side

Figure 24–10 and its accompanying text explain what happens on the supplier side during the multimaster replication process.

Figure 24–10 The Multimaster Replication Process on the Supplier Side



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* Advanced Replication immediately copies the change into the deferred transaction queue.

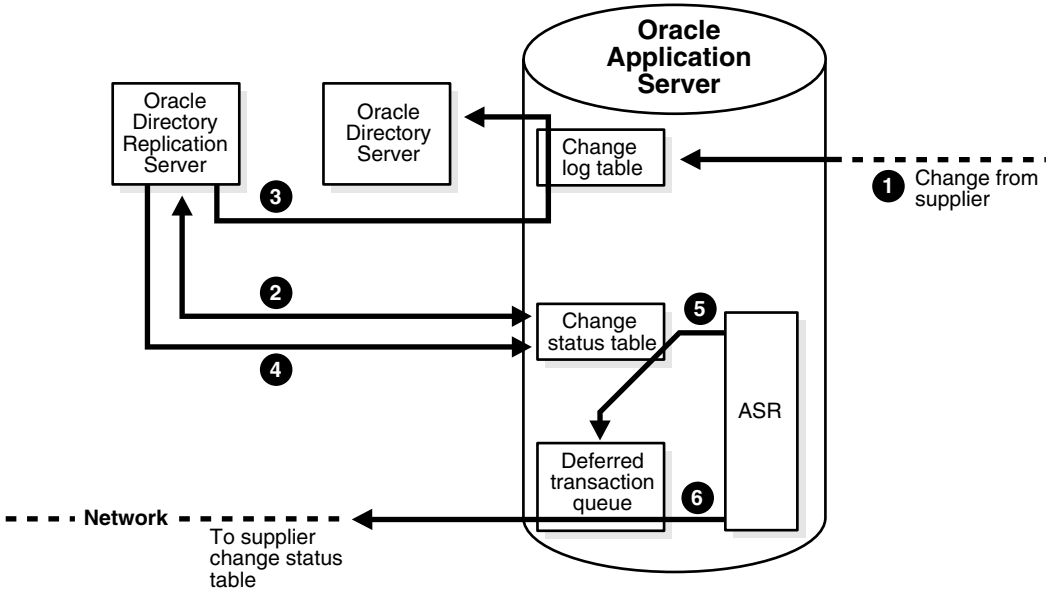
5. After a scheduled interval, Oracle9*i* Advanced Replication pushes pending transactions from the deferred transaction queue across the network to the consumer change log table.

> **Note:** All changes made to Oracle Application Server Single Sign-On tables by the single sign-on administrator application are also replicated by Oracle9*i* Advanced Replication.

The Multimaster Replication Process on the Consumer Side

Figure 24–11 and its accompanying text explain the multimaster replication process on the consumer side.

Figure 24–11 The Multimaster 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* Advanced Replication copies the change status update into the deferred transaction queue.
6. After the scheduled Oracle9*i* Advanced Replication interval, Oracle9*i* Advanced 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 those that are dropped as candidate changes. Remote change records in the local change log table are purged by the garbage collection thread if they have been applied locally. Local change records in the local change log table are purged by the garbage collection thread if they have been distributed to all the consumers.

> **See Also:** "Managing Replication" on page 25-35 for information on configuring replication

Conflict Resolution in Multimaster 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, for some reason, fails.

There are times when the replication process may not be able to apply a change. For example, suppose that Supplier Node A sends the consumer a change, and, immediately after that, Supplier Node B sends the consumer 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.

LDAP operations that can lead to conflicts include:

- Addition
- Deletion
- Modification
- Modification of either an RDN or a DN

Levels at Which Replication Conflicts Occur

There are two types of conflicts:

- Entry-level conflicts
- Attribute-level conflicts

Table 24–4 Types of Replication Conflict

Level of Replication Conflict	Description
Entry-level conflicts	An entry-level conflict is caused when the directory replication server attempts to apply a change to the consumer. One of the following types of changes to the consumer could occur: ■ 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.

Table 24–4 (Cont.) Types of Replication Conflict

Level of Replication Conflict	Description
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 differences in 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:

- Appendix B, "Oracle Internet Directory Schema Elements" for schema questions
- "The Catalog Management Tool (catalog.sh) Syntax" on page A-19 for catalog questions
- "Performing the Tasks of an Administrator" on page 31-10 for group entry questions

The Multimaster Replication Process

This section describes how the automated replication process adds, deletes, and modifies entries, and how it modifies DNs and RDNs.

How the Multimaster 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, then:

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*, then:

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, then:

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, then:

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, then:

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 Multimaster 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, then:

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*, then:

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, then:

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 Multimaster 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, then:

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*, then:

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, then:

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 Multimaster 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, then:

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, then:

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, then:

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, then:

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, then:

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 Multimaster 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, then:

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*, then:

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, then:

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, then:

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, then:

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.

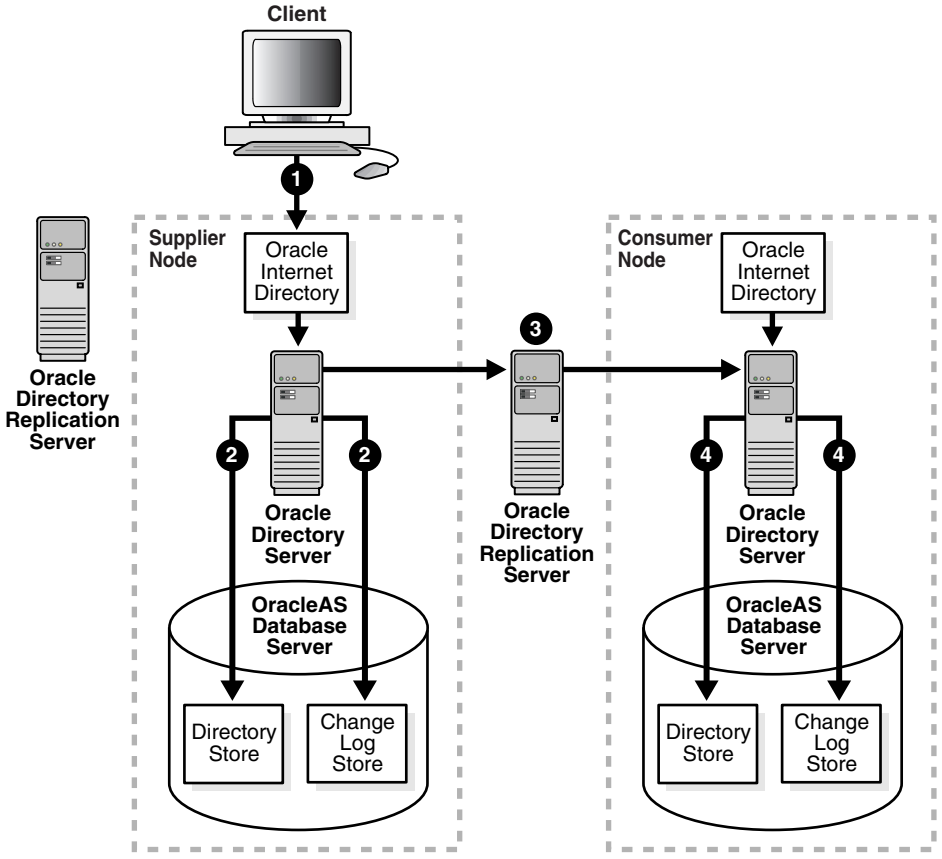
Fan-Out and Partial Replication

This section gives a more detailed look at fan-out and partial replication.

In fan-out replication, a consumer replicates data directly from a supplier. That consumer can then be a supplier to one or more other consumers.

Figure 24–12 and its accompanying text explain the fan-out replication process.

Figure 24–12 The Fan-Out Replication Process



As Figure 24–12 on page 24-34 shows:

1. An LDAP client issues a directory modification request to the directory on the supplier node.
2. The Oracle directory server on the supplier node performs the required modification in the directory store and simultaneously updates the change log store.

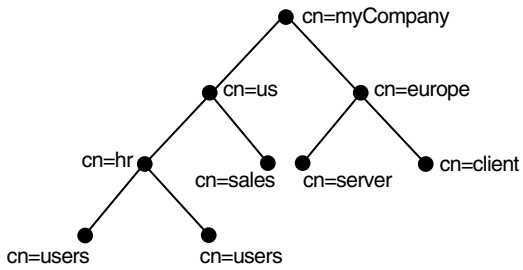
3. The directory replication server on the consumer node retrieves changes from the directory server on the supplier node and applies them to the directory server on the consumer.
4. The directory server on the consumer does the following simultaneously:
   - It makes the required modification, replicating the change in its directory store
   - It generates a shadow change log object in its change log store. The objects in this change log store can, in turn, be propagated to other fan-out consumers.
   - It updates the value of the `orcllastappliedchangenumber` attribute in the replication agreement entry to correspond to the number of the last change from the supplier node that it has applied

     See Also: "Change Logs in Directory Replication" on page 24-19 for more information about the `orcllastappliedchangenumber` attribute

Rules for Partial Replication Filtering

This section describes rules and best practices to follow when specifying naming contexts in partial replication. The discussion in this section relies on the sample naming context illustrated in Figure 24–13.

Figure 24–13 A Sample Naming Context



This section contains these topics:

- Scenario #1: The Included Naming Context in One Naming Context Object Is a Subtree of the Included Naming Context in Another Naming Context Object

Scenario #1: The Included Naming Context in One Naming Context Object Is a Subtree of the Included Naming Context in Another Naming Context Object

To use an example, suppose that you have a partial replication scenario in which the included naming context in Naming Context Object #2 is itself a subtree of the included naming context in Naming Context Object #1.

Naming Context Object #1

```
cn=namectx001,
cn=replication namecontext,
orclagreementid=unique_identifier_of_the_replication_agreement,
orclreplicaid=unique_identifier_of_the_supplier,
cn=replication configuration,
    orclincludednamingcontexts: cn=mycompany
```

Naming Context Object #2

```
cn=namectx002,
cn=replication namecontext,
orclagreementid=unique_identifier_of_the_replication_agreement,
orclreplicaid=unique_identifier_of_the_supplier,
cn=replication configuration
    orclincludednamingcontexts: cn=hr,c=us,cn=mycompany
    orclexcludednamingcontexts: cn=users,cn=hr,c=us,cn=mycompany
    orclexcludedattributes: userpassword
```

In this scenario, the naming context that is replicated is the highest one specified in the `orclincludednamingcontexts` attribute. Any excluded naming contexts are not replicated. To use our example, all changes under the subtree `cn=mycompany`—except `cn=users,cn=hr,c=us,cn=mycompany`—are replicated. The attribute `userpassword` is excluded from all changes replicated under `cn=hr,c=us,cn=mycompany`.

Scenario #2: The Included Naming Context in One Naming Context Object Is a Subtree of An Excluded Naming Context in Another Naming Context Object

To use an example, suppose that you have a partial replication scenario in which the excluded naming context in Naming Context Object #4 is a subtree of the excluded naming context defined in Naming Context Object #3.

Naming Context Object #3

```
cn=namectx001,cn=replication namecontext,
orclagreementid=identifier,orclreplicaid=supplier,cn=replication configuration
        orclincludednamingcontexts: cn=mycompany
        orclexcludednamingcontexts: cn=us,cn=mycompany
```

Naming Context Object #4

```
cn=namectx002,cn=replication
namecontext,orclagreementid=identifier,orclreplicaid=supplier,cn=replication
configuration
        orclincludednamingcontexts: cn=hr, c=us,cn=mycompany
        orclexcludednamingcontexts: cn=users,cn=hr,c=us,cn=mycompany
        orclexcludedattributes: userpassword
```

In this scenario, the included naming context specified in Naming Context Object #4 is not replicated. That naming context is a subtree of a specified excluded naming context in Naming Context Object #3. In this case, Naming Context Object #4 is ignored, and no changes under `cn=hr,c=us,cn=mycompany` are replicated.

Rules for Managing Naming Contexts and Attributes

Reserved Naming Contexts and Attributes

The following naming contexts cannot be replicated:

```
orclagreementid=000001,cn=replication configuration
cn=subconfigsubentry
cn=Oracle Internet Directory
cn=subregistrysubentry
```

The following naming contexts cannot be excluded from replication:

```
cn=catalogs
cn=subschemasubentry
cn=orclshemaversion
```

```
cn=replication configuration
```

The following attributes cannot be excluded from replication whether they are mandatory or optional.

```
orclguid
creatorsname
createtimestamp
cn
dn
attributetypes
objectclasses
objectclass
orclindexedattribute
orclproductversion
```

Other Attributes That Cannot Be Excluded from Replication

You cannot exclude mandatory attributes from replication. For example, suppose that you have an object class named `my_object_class`, which includes the following attributes: `mandatory_attribute_1`, `optional_attribute_1`, and `optional_attribute_2`. In this case, you cannot exclude from replication `mandatory_attribute_1`.

Naming attributes are always replicated even if you specify them for exclusion from replication.

Optimization of Partial Replication for Better Performance

If not carefully planned, partial replication can degrade performance of the replication process. For example, the definition of naming contexts in Naming Context Object #5 decreases performance as compared to that in Naming Context Object #6.

Naming Context Object #5

In this naming context object, there are two naming context objects in partial replication.

```
cn=namectx001,cn=replication
namecontext,orclagreementid=identifier,orclreplicaid=supplier,cn=replication
configuration
orclincludednamingcontexts: cn=mycompany
orclexcludednamingcontexts: c=europe,cn=mycompany
orclexcludedattributes: userpassword
```

Naming Context Object #6

```
cn=namectx002,cn=replication
namecontext,orclagreementid=<id>,orclreplicaid=<supplier>,cn=replication
configuration
orclincludednamingcontexts: cn=hr, c=us,cn=mycompany
orclexcludednamingcontexts: cn=users,cn=hr, c=us,cn=mycompany
orclexcludedattributes: userpassword
```

If these two naming context objects are defined, then all changes under `cn=mycompany` are replicated—except `cn=europe,c=mycompany` and `cn=users,cn=hr,c=us,cn=mycompany`. The attribute `userpassword` is filtered out. However, as you can see in Naming Context Object #7, you can develop a single naming context object that fulfills the same requirement, and that avoids unnecessary performance degradation in partial replication.

Naming Context Object #7

```
cn=namectx001,cn=replication
namecontext,orclagreementid=identifier,orclreplicaid=supplier,cn=replication
configuration
orclincludednamingcontexts: cn=mycompany
orclexcludednamingcontexts: c=europe,cn=mycompany
orclexcludednamingcontexts: cn=users,cn=hr, c=us,cn=mycompany
orclexcludedattributes: userpassword
```

25

Oracle Directory Replication 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.

This chapter contains these topics:

- Installing and Configuring Multimaster Replication
- Installing and Configuring LDAP-Based Replication
- Managing Replication
- Example: Installing and Configuring a Multimaster Replication Group with Fan-Out

> **See Also:** "Directory Replication" on page 2-23 for a conceptual discussion of replication

Installing and Configuring Multimaster Replication

This section tells you how to install and configure multimaster replication groups, and how to resolve conflicts manually in them. It contains these topics:

Installing and Configuring a Multimaster Replication Group

This section discusses the general tasks you perform when installing and configuring a multimaster replication group. It contains these topics:

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 Node to a Multimaster Replication Group" on page 25-13.
- In Oracle Internet Directory 10*g* (9.0.4), a node cannot be part of more than one multimaster replication group.
- 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.
- The data replicated between servers in a directory replication group does not include DSE root-specific data, server configuration data, and replication agreement data.
- When an Oracle Internet Directory multimaster replication group is configured, the Oracle Application Server Single Sign-On database schema is automatically configured in replication.

Preliminary Information for Installing and Configuring a Multimaster Replication Group

This section describes the types of installation you need to perform to configure a multimaster replication group. It also introduces the Replication Environment Management Tool that enables you to perform various configuration tasks.

Oracle9*i* Enterprise Edition In Oracle Internet Directory 10*g* (9.0.4), performing multimaster replication requires **Oracle9i Advanced Replication**, which is part of a typical installation of the Oracle9*i* Enterprise Edition. A typical installation of Oracle9*i* Standard Edition does not include Oracle9*i* Advanced Replication.

Oracle Application Server Infrastructure When you install Oracle Internet Directory as part of Oracle Application Server on any node, you are prompted to select a product. Choose the Oracle Application Server Infrastructure. Then, later in the installation process, you are prompted to choose one of various installation types. The installation type that you must choose depends on whether you are installing

on a node that serves as a **master definition site (MDS)** or one that serves as a **remote master site (RMS)**.

If you are installing on a Master Definition Site

1. Choose the "Identity Management and Oracle Application Server Metadata Repository" installation type. Choose Next. The Select Configuration Options screen appears.
2. Verify that all options are selected.
3. Choose Next.

If you are installing on a Remote Master Site

1. Choose "Identity Management and Oracle Application Server Metadata Repository" installation type. Choose Next. The Select Configuration Options screen appears.
2. Deselect everything in the Select Configuration Options screen.
3. Choose Next.

Later, the Oracle Universal Installer asks for the host and port of Oracle Internet Directory. Specify the host and port number of the MDS. Verify that the server is running on that node.

After installation, create the wallet by entering the following:

```
$Oracle_Home/bin/Oidpasswd connect=connect_string  create_wallet=TRUE
current_password=password_for_the_ODS_database_user
```

Start and shut down the Oracle Internet Directory processes by entering the following:

```
$Oracle_Home/bin/oidmon connect=connect_string start

$Oracle_Home/bin/oidctl connect=connect_string server=oidldapd instance=1 start

$Oracle_Home/bin/oidmon connect=connect_string stop
```

The Replication Environment Management Tool During installation and configuration, you use the Replication Environment Management Tool to perform various tasks. This tool assists you in:

- Configuring a replication group
- Adding and deleting replicas

- Managing the directory replication group
- Modifying or resetting the replication Bind BN password
- Modifying the database replication user REPADMIN password
- Displaying various errors and status information for change log propagation

---

Note: You do not need Oracle9*i* Advanced Replication to perform partial—that is, LDAP-based—replication.

In a directory replication group, nodes can have different patchset versions of the Oracle9*i* Database Server if they have the same version of Oracle Internet Directory.

If the nodes in a directory replication group are running different versions of Oracle Internet Directory, you can modify directory servers on those nodes. However, do not replicate changes generated on a newer version of Oracle Internet Directory to a node that has not yet upgraded to that version. Otherwise, the changes can contain information that the earlier version cannot properly interpret.

See Also: "The Replication Environment Management Tool" on page A-62 for more information about the Replication Environment Management Tool

Task 1: Install Oracle Internet Directory on the Master Definition Site

You must be able to use **Oracle Net Services** to connect to the master definition site database and all other nodes in the DRG.

Note: During installation, be sure that each Oracle Internet Directory database instance name is unique on each machine.

See Also:

- "Oracle Application Server Infrastructure" on page 25-3 for instructions on installing on a master definition site
- Installation documentation for Oracle Internet Directory

Task 2: Install the Oracle Internet Directory on the Remote Master Sites

> **See Also:** "Oracle Application Server Infrastructure" on page 25-3 for instructions on installing on a remote master site

Task 3: Set Up Oracle9*i* Advanced Replication for a Directory Replication Group

The following sections lead you through installing and configuring Oracle9*i* Advanced Replication through Oracle Internet Directory installation scripts. More advanced Oracle9*i* Advanced Replication users may prefer to configure Oracle9*i* Advanced Replication through the Oracle9*i* Advanced Replication Manager Tool.

> **See Also:** *Oracle9i Advanced Replication* in the Oracle Database Documentation Library, and the online Help for Oracle9*i* Advanced Replication Manager, for information on configuring Oracle9*i* Advanced Replication by using the Oracle9*i* Advanced Replication Manager

To configure the Oracle9*i* Advanced 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 Advanced Replication For Directory Replication

On All Nodes, Prepare the Oracle Net Services Environment for Replication To prepare the Oracle Net Services environment, follow these steps, described more fully in this section, on *all nodes* in the directory replication group:

1. Configure `sqlnet.ora`.
2. Configure `tnsnames.ora`.
3. Optional: Create rollback table space and rollback segments.
4. If you created rollback table space and rollback segments, then modify the parameters in the initialization parameter file, `init.ora`.
5. Stop and restart the listener.
6. IF you created rollback table space and rollback segments, then stop and restart the Oracle Internet Directory database.
7. IMPORTANT: 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:

   ```
   connect_string =
       (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. If you created rollback table space and rollback segments, then 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. IF you created rollback table space and rollback segments, then 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. **IMPORTANT:** 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* Advanced Replication For Directory Replication To do this:

1. Connect as the system user on all nodes, including the MDS, from the MDS console. Ensure the following on all nodes:
   - The Oracle Internet Directory database is running
   - The Oracle Internet Directory listener is running
   - The connect string is correct
   - The system password is correct
2. Ensure the following on remote sites:
   - A wallet exists for storing the password to the database designated for Oracle Internet Directory. This wallet is named `oidpwdl` and is located in the directory *`ORACLE_HOME`*`/ldap/admin`.
   - A wallet exists for storing the password of the replication administrator. This wallet is named `oidpwdr`, and is located in the directory *`ORACLE_HOME`*`/ldap/admin`.

3. From the MDS, at the command prompt, run the following script if the prerequisites in the following note are met:

   ```
   ORACLE_HOME/ldap/bin/remtool -asrsetup
   ```

   The Replication Environment Management Tool (remtool) configures Oracle9*i* Advanced Replication.

   Note: If you encounter errors, then clean up the environment by using the `-asrcleanup` option of the Replication Environment Management Tool. Then repeat Step 3.

   See Also:

   - "-ASRSETUP Option" on page A-68 for instructions on using the -ASRSETUP option of the Replication Environment Management Tool (remtool) and an example
   - *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
   - The chapter on Oracle Wallet Manager in *Oracle Advanced Security Administrator's Guide* for instructions on creating an Oracle wallet

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 connect_string -check -generate file_with_absolute_
   path_name
   ```

Note: If data is extracted from Oracle Internet Directory, then, in addition to other options, use the `-restore` option to restore the operational attributes.

2. From the same node, enter:

   ```
   bulkload.sh -connect connect_string_1 -load
   ```

3. Repeat Step 2, each time replacing *`connect_string_1`* with the connect string of another node in the DRG, until you have loaded the data onto all the nodes. For example, enter:

   ```
   bulkload.sh -connect connect_string_2 -load
   ```

   then enter:

   ```
   bulkload.sh -connect connect_string_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 1.3.2.2-1 or later. Visit:
  `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit:
  `http://www.datafocus.com/`

See Also:

- See "bulkload Syntax" on page A-45 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 commands on each node:

```
oidmon [connect=connect_string] [sleep=seconds] start
oidctl connect=connect_string server=oidldapd instance=instance_number_of_
directory_server flags='-h host_name -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_directory_server` need not be unique across the entire DRG. For example, you can have `instance=1` on both node A and 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 on each node:

```
oidctl connect=connect_string server=oidrepld instance=1
   flags='-h host_on_which_the_directory_server_is_running -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 replication 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 Multimaster Replication" on page 24-24

Note: As part of Task 3, the Replication Environment Management Tool (`remtool`) sets normal defaults enabling you to simply start the replication servers. If you wish to alter these defaults, see Managing Replication on page 25-35.

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 in succession 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.

Note: If you want to configure replication for Oracle Application Server Single Sign-On, then follow the post-installation steps specific to Oracle Application Server Single Sign-On. These are found in the replication installation section of the *Oracle Application Server Single Sign-On Administrator's Guide*.

Adding a Node to a Multimaster Replication Group

Note: A new node that you add to an existing multimaster replication group must have Oracle Application Server Infrastructure product installed on it. During that installation, the installation type must have been "Oracle Application Server Metadata Repository". For more information, see "Task 2: Install the Oracle Internet Directory on the Remote Master Sites" on page 25-6.

There are two ways to add a new node to a live replication group.

- Using ldifwrite and bulkload

  If your directory contains less than one million entries, then use this method.

  This method involves using the ldifwrite utility to back up LDAP data with operational attributes preserved. Once this is done, the bulkload utility is then used to load data to all replicas in a group.

  Use bulkload with the `-check`, `-generate`, and `-restore` arguments once, and then with the `-load` argument once for each replica. When using the

`-load` argument on each replica, preserve the operational attributes by using the same intermediate files generated by using the `-generate` argument.

Backup using this method can take up to seven hours for a directory with one million entries.

- Using cold backup

  For a directory of more than a million entries, this method takes much less time than the previously mentioned method.

  See Also: Appendix F, "Addition of a Directory Node by Using the Database Copy Procedure"

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 25-6.

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 Advanced 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 the Directory 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 the Master Definition Site", 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
```

This switches all running Oracle directory servers to read-only mode.

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 Advanced Replication Add Node Setup" while backup is in process.

Enter the following command:

```
ldifwrite -c connect string -b "orclAgreementID=000001,cn=replication
configuration" -f output_ldif_file
```

Task 4: Perform Oracle9*i* Advanced 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".

On the sponsor node, enter this command:

```
ORACLE_HOME/ldap/bin/remtool -addnode
```

The Replication Environment Management Tool adds the node to the DRG.

Note: If you encounter errors, then use the `-asrverify` option first. If it reports errors, then rectify them by using the `-asrrectify` option. Both `-asrverify` and `-asrrectify` list all nodes in the DRG. If the new node is not in the list, then add it by running the Replication Environment Management tool again, using the `-addnode` option.

See Also: "-ADDNODE Option" on page A-65 of the Replication Environment Management Tool for instructions on using the -ADDNODE option and an example

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
```

This switches all running Oracle directory servers to read/write mode.

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 -check -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 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

Task 8: Start the Directory Server on the New Node

To start the directory 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 25-35.

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
```

> **Note:** Once a directory server instance is participating in a replication agreement, do not use the bulkload tool to add data into the node. Instead, use ldapadd.
>
> If Oracle Application Server Single Sign-On is desired in replication, then follow the *Oracle Application Server Single Sign-On Administrator's Guide* in the replication installation section for the post-installation steps specific to Oracle Application Server Single Sign-On.

Deleting a Node from a Multimaster Replication Group

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.

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, perform these tasks, each of which is more fully described in this section.

Task 1: Stop the Directory Replication Server on All Nodes

Task 2: Stop All Processes in the Node to be Deleted

Task 3: Delete the Node from the Master Definition Site

Task 4: Start the Directory Replication Server on All Nodes

Task 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=connect_string server=oidrepld instance=1 stop
```

Note: The instance number may vary.

Task 2: Stop All Processes in the Node to be Deleted

On the node to be deleted, stop the **OID Monitor** and all directory server instances.

```
oidmon [connect=connect_string] [host=virtual/host_name] stop
oidctl connect=connect_string server=oidldapd instance=server_instance_number
stop
```

See Also:

- "Stopping the OID Monitor" on page A-5 for instructions about stopping the OID Monitor
- "Stopping an Oracle Directory Server Instance" on page A-9 for instructions about stopping a directory server instance by using the OID Control Utility

Task 3: Delete the Node from the Master Definition Site

From the **MDS**, run the following script:

```
remtool -delnode
```

The Replication Environment Management Tool deletes the node from the replication group.

See Also: "-DELNODE Option" on page A-73 of the Replication Environment Management Tool for instructions on using the -DELNODE option and an example

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 you encounter errors, then use the `-asrverify` option first. If it reports errors, then rectify them by using the `-asrrectify` option. Both `-asrverify` and `-asrrectify` list all nodes in the DRG. If the node to be deleted is not in the list, then add it by running the Replication Environment Management tool again, using the `-delnode` option.

Task 4: Start the Directory Replication Server on All Nodes

To start the directory replication server, type the following command:

```
oidctl connect=connect_string server=oidrepld instance=1
flags='-h host -p port' start
```

See Also: "Starting an Oracle Directory Replication Server Instance" on page A-9

Resolving Conflicts Manually in a Multimaster Replication Group

This section contains these topics:

- Monitoring Replication Change Conflicts
- Examples of Conflict Resolution Messages
- About the Human Intervention Queue Manipulation Tool
- About 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 in this section, 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.
```

About the Human Intervention Queue Manipulation Tool

The Human Intervention Queue Manipulation Tool enables you to move 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. To address changes in the human intervention queue, follow these general steps:

1. Shut down 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:** "The Human Intervention Queue Manipulation Tool" on page A-56 for instructions on how to use the Human Intervention Queue Manipulation Tool

About 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 a tranquil state—that is, that neither is supplying or applying changes. 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:**
>
> - "Task 2: Identify a Sponsor Node and Switch the Sponsor Node to Read-Only Mode" for instructions on setting a node to read-only mode
> - "The OID Reconciliation Tool" on page A-59 for syntax and an explanation of how OID Reconciliation Tool works.

Installing and Configuring LDAP-Based Replication

This section contains these topics:

- Rules for Configuring LDAP-Based Replication
- Installing an LDAP-Based Replica
- Configuring an LDAP-Based Replica
- Deleting an LDAP-Based Replica

- Determining What Is to Be Replicated in LDAP-Based Partial Replication

Rules for Configuring LDAP-Based Replication

The following rules apply to both full and partial LDAP-based replication:

- An LDAP-based replica cannot have two suppliers
- In LDAP-based replication, only the naming contexts listed in the `namingcontexts` attribute of the root DSE can be replicated to the consumer.
- The supplier of an LDAP-based replica can be either a standalone node or a member of a multimaster replication group.
- An LDAP-based replica can be a consumer for another LDAP-based replica. It is then called a fan-out replica.

> **See Also:** For instructions on installing on a standalone node, see "If you are installing on a Master Definition Site" on page 25-4

Installing an LDAP-Based Replica

When you install Oracle Internet Directory on any given node, follow these steps:

1. When you are prompted to select a product, choose the **Oracle Application Server Infrastructure**.
2. Choose the **Identity Management and Oracle Application Server Metadata Repository** installation type. Choose **Next**. The Select Configuration Options screen appears.
3. Deselect everything in the Select Configuration Options screen.
4. Choose **Next**.

Later, the Oracle Universal Installer asks for the host and port of Oracle Internet Directory. Specify the host and port number of the supplier. Verify that the server is running on that node.

After installation, create the wallet by entering the following:

```
$ORACLE_HOME/bin/Oidpasswd connect=connect_string  create_wallet=TRUE
current_password=password_for_the_ODS_database_user
```

Start and shut down the Oracle Internet Directory processes by entering the following:

```
$ORACLE_HOME/bin/oidmon connect=connect_string start
```

```
$ORACLE_HOME/bin/oidctl connect=connect_string server=oidldapd instance=1 start

$ORACLE_HOME/bin/oidmon connect=connect_string stop
```

Configuring an LDAP-Based Replica

How you configure an LDAP-based replica depends on whether you have backed up the directory by using the ldifwrite tool or by using automatic bootstrapping. Table 25–1 compares these two methods.

Table 25–1 A Comparison of Backup and Automatic Bootstrapping

Backup Using Idifwrite	Automatic Bootstrapping
Manual procedure	Automatic procedure
Faster performance	Uses the filtering capability of partial replication
Good for a large amount of data	Good for a smaller number of entries

Configuring an LDAP-Based Replica by Using Automatic Bootstrapping

This section discuss the general tasks you perform when configuring an LDAP-based replica by using automatic bootstrapping. It contains these topics:

- Task 1: Identify the Supplier Node
- Task 2: Add an LDAP-Based Replica by Using the Replication Environment Management Tool
- Task 3: Configure the Replica for Automatic Bootstrapping
- Task 4: Optional: Change Default Replication Parameters
- Task 5: Start the Directory Replication Server on the Consumer Replica

Task 1: Identify the Supplier Node Identify the supplier for an LDAP-based replica. The supplier can be:

- A standalone directory
- A node of a multimaster replication group
- Another LDAP-based replica

Task 2: Add an LDAP-Based Replica by Using the Replication Environment Management Tool To add a replica, enter the following:

```
remtool -paddnode [-v] [-bind supplier_host_name:port/replication_dn_password]
```

> **See Also:** "The Replication Environment Management Tool" on page A-62 for more information about the Replication Environment Management Tool

Task 3: Configure the Replica for Automatic Bootstrapping To use the automatic bootstrap capability, set the `orclReplicaState` attribute of the replica subentry to `0` as follows:

1. Edit the sample file `mod.ldif` as follows:

```
Dn: orclreplicaid=<unique replica identifier>, cn=replication configuration
Changetype:modify
add:orclReplicaState
OrclReplicaState: 0
```

2. Use ldapmodify to update the replica subentry `orclreplicastate` attribute.

```
Ldapmodify -D "cn=orcladmin" -w administrator_password -h my_host -p 389 -f
mod.ldif
```

> **See Also:** "Managing Replication" on page 25-35 for more information about the bootstrap capability of the LDAP-based replication

Task 4: Optional: Change Default Replication Parameters You can change the default parameters for replication agreements and for the replica subentry.

See Also:

- "Viewing and Modifying Directory Replication Server Configuration Parameters" on page 25-36
- "Viewing and Modifying Parameters for Particular Replica Nodes" on page 25-38
- "Modifying Parameters for Replication Agreements" on page 25-40
- "Replication Configuration Objects in the Directory" on page 24-13
- "Determining What Is to Be Replicated in LDAP-Based Partial Replication" on page 25-31

Task 5: Start the Directory Replication Server on the Consumer Replica

See Also: "Starting an Oracle Directory Replication Server Instance" on page A-9

Configuring an LDAP-Based Replica by Using the ldifwrite Tool

This section discuss the general tasks you perform when configuring an LDAP-based replica by using the ldifwrite tool. It contains these topics:

- Task 1: Start the Directory Server on Both the Supplier and the Consumer Nodes
- Task 2: Change the Directory Server at the Supplier to Read-Only Mode
- Task 5: Back Up the Naming Contexts to Be Replicated
- Task 3: Add an LDAP-Based Replica by Using the Replication Environment Management Tool
- Task 4: Initialize the lastappliedchangenumber Attribute
- Task 6: Change the Directory Server at the Supplier to Read/Write Mode
- Task 7: Load the Data on the LDAP-Based Replica
- Task 8: Optional: Change Default Replication Parameters
- Task 9: Start the Directory Replication Server on the Consumer Replica

Task 1: Start the Directory Server on Both the Supplier and the Consumer Nodes Identify the supplier for an LDAP-based replica, and verify that the directory server is running on both the supplier and the consumer. The supplier can be:

- A standalone directory
- A node of a multimaster replication group
- Another LDAP-based replica

Task 2: Change the Directory Server at the Supplier to Read-Only Mode To ensure data consistency, change the directory server on the supplier node to read-only. To do this:

1. Create an LDIF file containing the following:

```
Dn:
Changetype: modify
Replace: orclservermode
Orclservermode: r
```

2. On the supplier, run the following command:

```
ldapmodify -D "cn=orcladmin" -w administrator_password -h host_name_of_
supplier_node -p port -f name_of_LDIF_file.ldif
```

Task 3: Add an LDAP-Based Replica by Using the Replication Environment Management Tool To add a replica, enter the following:

```
remtool -paddnode [-v] [-bind supplier_host_name:port/replication_dn_password]
```

> **See Also:** "The Replication Environment Management Tool" on page A-62 for more information about the Replication Environment Management Tool

Task 4: Initialize the lastappliedchangenumber Attribute To do this:

1. Search for the last applied change number on the supplier node.

```
ldapsearch -D bind_DN -w password -h host_name -p port_number -b "" -s base
"objectclass=*" lastchangenumber
```

2. Modify the corresponding agreement with the retrieved last applied number at the supplier. To do this:

   a. On the supplier, create an LDIF file with the retrieved last applied change number:

```
dn:orclagreementid=agreement_identifier,orclreplicaid=supplier_replica_
identifier,cn=replication configuration
changetype: modify
replace: orclLastAppliedChangeNumber
orclLastAppliedChangeNumber: last_change_number_retrieved_in_step_1.
```

b. Modify the agreement by using ldapmodify:

```
ldapmodify -D bind_DN -w password -h host_name -p port_number -f LDIF_
file
```

Task 5: Back Up the Naming Contexts to Be Replicated If there is a large number of entries in the naming contexts that you want to replicate to the LDAP-based replica, then Oracle Corporation recommends that you back up these naming contexts at the supplier node and then load them to the LDAP-based replica.

To back up the naming contexts:

1. Identify the replication agreement DN created in "Task 3: Add an LDAP-Based Replica by Using the Replication Environment Management Tool" on page 25-27.

   ```
   ldapsearch -h supplier host -p port number -b "orclreplicaid=supplier
   replica ID, cn=replication configuration" -s sub "(orclreplicadn=
   orclreplicaid=consumer replica ID, cn=replication configuration)" dn
   ```

2. Use the following command to get the data from the supplier. Data loaded into the file will be based on the agreement configured:

   ```
   ldifwrite -c connect string of sponsor node -b "replication agreement dn"
   -f name of output LDIF file.ldif
   ```

Note: You might want to perform "Task 8: Optional: Change Default Replication Parameters" on page 25-30 before backing up the data so that additional changes in the agreement are taken care of during the backup.

See Also:

"Determining What Is to Be Replicated in LDAP-Based Partial Replication" on page 25-31

"Example 2: Converting Part of a Specified Naming Context to an LDIF File" on page A-55 for more instructions on using ldifwrite to back up part of the naming context

Task 6: Change the Directory Server at the Supplier to Read/Write Mode If you performed "Task 2: Change the Directory Server at the Supplier to Read-Only Mode" on page 25-27, then change the directory server on the supplier back to read/write mode. To do this:

1. Create an LDIF file containing the following:

```
Dn:
Changetype: modify
Replace: orclservermode
Orclservermode: rw
```

2. On the supplier, run the following command:

```
ldapmodify -D "cn=orcladmin" -w administrator_password -h host_name_of_
supplier_node -p port -f name_of_LDIF_file.ldif
```

Task 7: Load the Data on the LDAP-Based Replica To do this:

1. If there are multiple files, then combine them into one file—for example, `backup_data.ldif`.
2. If naming contexts exist on the LDAP-based consumer replica, then remove them by using bulkdelete. Enter the following:

```
Bulkdelete.sh -connect connect_string_of_replica  -b "naming_context"
```

Perform this step for each naming context that was backed up in "Task 5: Back Up the Naming Contexts to Be Replicated" on page 25-28.

Load the data to the replica by using bulkload in the append mode. Enter the following:

```
Bulkload.sh -connect connect_string_of_replica -append -check -generate -load
-restore backup_data.ldif
```

See Also: "bulkload Syntax" on page A-45 for instructions on using bulkload in either the default mode or the append mode

Task 8: Optional: Change Default Replication Parameters You can change the default parameters for replication agreements and for the replica subentry.

See Also:

- "Viewing and Modifying Directory Replication Server Configuration Parameters" on page 25-36
- "Viewing and Modifying Parameters for Particular Replica Nodes" on page 25-38
- "Modifying Parameters for Replication Agreements" on page 25-40
- "Replication Configuration Objects in the Directory" on page 24-13
- "Determining What Is to Be Replicated in LDAP-Based Partial Replication" on page 25-31

Task 9: Start the Directory Replication Server on the Consumer Replica

See Also: "Starting an Oracle Directory Replication Server Instance" on page A-9

Deleting an LDAP-Based Replica

This section explains how to delete an LDAP-based replica. It contains these topics:

- Task 1: Stop the Directory Replication Server on the Node to be Deleted
- Task 2: Stop the Directory Server on the Node to be Deleted
- Task 3: Delete the Replica from the Replication Group

Note: You cannot delete a replica if it is a supplier for another replica. To delete such a replica, you must first delete all its consumers from the replication group.

Task 1: Stop the Directory Replication Server on the Node to be Deleted

See Also: "Stopping an Oracle Directory Replication Server Instance" on page A-11

Task 2: Stop the Directory Server on the Node to be Deleted

See Also: "Stopping an Oracle Directory Server Instance" on page A-9

Task 3: Delete the Replica from the Replication Group

Do this by using the Replication Environment Management Tool. Enter:

```
remtool -pdelnode [-v] [-bind hostname:port_number/replication_dn_password]
```

Determining What Is to Be Replicated in LDAP-Based Partial Replication

See Also: "The Replication Environment Management Tool" on page A-62

In LDAP-based partial replication, you can determine what is or is not replicated by defining replica naming context objects. The parameters for these objects are stored in entries that have this DN:

```
cn=namingcontext_ID,cn=replication namecontext,
orclAgreementID=numeric_identifier_of_replication_agreement,
orclReplicaId=unique_identifier_of_replica, cn=replication configuration
```

Note: Because the directory replication server reads replica naming context objects from the agreement located at the supplier, you must apply all modifications against naming context objects at the supplier and, optionally, at the consumer.

Viewing and Modifying Replica Naming Context Objects by Using Oracle Directory Manager

To view and modify parameters for replica naming context objects:

1. In the navigator pane, expand in succession **Oracle Internet Directory Servers**, ***directory server instance***, **Replication Management**, **Replica Node:** ***replica identifier***, **Replica Agreement:** ***replication agreement identifier***
2. Select the replica naming context you want to modify. The **Replica Naming Context** tab page appears in the right pane. The fields in this tab page are described in Table C–18 on page C-16.
3. After you have entered the appropriate information, choose **OK**.

Adding Replica Naming Context Objects by Using Oracle Directory Manager

1. In the navigator pane, expand in succession **Oracle Internet Directory Servers**, ***directory server instance***, **Replication Management**, **Replica Node:** ***replica identifier***, **Replica Agreement:** ***replication agreement identifier***.
2. Select **Naming Context:***naming context identifier*.
3. From the toolbar, choose **Create**. The New Replica Agreement Naming Context dialog box appears.
4. In the fields in the New Replica Agreement Naming Context dialog box, enter the appropriate information. The fields in this dialog box are described in Table C–18 on page C-16.
5. Choose **OK**.

Deleting Replica Naming Context Objects by Using Oracle Directory Manager

1. In the navigator pane, expand in succession **Oracle Internet Directory Servers**, ***directory server instance***, **Replication Management**, **Replica Node:** ***replica identifier***, **Replica Agreement:** ***replication agreement identifier***.
2. Using your mouse, right-click **Naming Context:***naming context identifier*.
3. Select **Delete**.

Modifying Replica Naming Context Object Parameters by Using ldapmodify

Replica naming context object parameters are listed and described in Table B–33 on page B-40.

Example 1: Adding a Naming Context Object for an LDAP-Based Replica

This example creates a naming context object that does the following:

- Replicates the naming context `ou=Americas,cn=mycompany`
- Excludes from replication the naming context `cn=customer profile, ou=Americas, cn=mycompany`
- Excludes from replication the attribute `userpassword`

The steps are:

1. Edit the example file `mod.ldif` as follows:

```
dn: cn=naming_context_identifier,cn=replication
namecontext,orclagreementid=replication_agreement_
identifier,orclreplicaid=consumer_replica_identifier,cn=replication
```

```
configuration
orclincludednamingcontexts: ou=Americas,cn=mycompany
orclexcludednamingcontexts: cn=customer profile, ou=Americas, cn=mycompany
orclexcludedattributes: userpassword
objectclass: top
objectclass: orclreplnamectxconfig
```

2. Use ldapadd to add the partial replication naming context object to both the supplier and the consumer.

```
ldapadd -D "bind_DN" -w administrator_password -h host -p port_number -f
mod.ldif
```

Example 2: Deleting a Naming Context Object for an LDAP-Based Replica

This example deletes from both the supplier and the consumer the naming context object created in the previous example.

The command is:

```
ldapdelete -D "bind_DN" -w administrator_password
-h [supplier host | consumer host]
-p port_number
"cn=naming_context_identifier,
cn=replication namecontext,orclagreementid=replication_agreement_
identifier,orclreplicaid=consumer_replica_identifier,cn=replication
configuration"
```

Example 3: Modifying the orclIncludedNamingcontexts Attribute for a Replica Naming Context Object

The directory replication server uses the `orclIncludedNamingcontexts` attribute value of the replica naming context object to specify the top-level subtree included in partial replication.

In this example, the included naming context is set to `c=us`, which means that `cn=us` is to be included in partial replication.

1. Edit the example file `mod.ldif` as follows:

```
cn=naming_context_identifier,
cn=replication namecontext,
orclagreementid=replication_agreement_identifier,
orclreplicaid=consumer_replica_identifier,
cn=replication configuration
Changetype:modify
```

```
Replace: orclIncludedNamingcontexts
orclIncludedNamingcontexts: c=us
```

2. Use ldapmodify to update the replication agreement orclupdateschedule attribute.

```
ldapmodify -D "cn=orcladmin" -w administrator_password -h my_host -p port -f
mod.ldif
```

3. Restart the directory replication server.

Example 4: Modifying the orclExcludedNamingcontexts Attribute for a Replica Naming Context Object

The directory replication server uses the `orclExcludedNamingcontexts` attribute value of the replica naming context object to specify the top-level subtrees excluded from partial replication.

In this example, the excluded naming contexts are set to `ou=Europe,c=us` and `ou=Americas,c=us`, which means that these two naming contexts are to be excluded from partial replication.

1. Edit the example file `mod.ldif` as follows:

```
cn=naming_context_identifier,
cn=replication namecontext,orclagreementid=replication_agreement_
identifier,orclreplicaid=consumer_replica_identifier,cn=replication
configuration
Changetype:modify
Replace: orclExcludedNamingcontexts
orclExcludedNamingcontexts: ou=Europe, c=us
orclExcludedNamingcontexts: ou=Americas, c=us
```

2. Use ldapmodify to update the replication agreement orclupdateschedule attribute.

```
ldapmodify -D "cn=orcladmin" -w administrator_password -h my_host -p port -f
mod.ldif
```

3. Restart the directory replication server.

Note: A subtree specified in the `orclexcludednamingcontexts` attribute must also be a subtree of the specified `includednamingcontext` of the same replica naming context object.

Example 5: Modifying the orclExcludedAttributes Attribute for a Replica Naming Context Object

You can specify that certain changes made to the included naming context be excluded from partial replication. To determine which attributes are to be excluded, the directory replication server uses the value of the `orclExcludedAttributes` attribute of the replica naming context object.

In this example, the `telephonenumber` and `title` attributes of the naming context specified in the orclincludednamingcontexts attribute are excluded from replication.

1. Edit the example file `mod.ldif` as follows:

```
cn=naming_context_identifier,
cn=replication namecontext,orclagreementid=replication_agreement_
identifier,orclreplicaid=consumer_replica_identifier,cn=replication
configuration
Changetype:modify
Replace: orclExcludedAttributes
orclExcludedAttributes: telephonenumber
orclExcludedAttributes: title
```

2. Use ldapmodify to update the replication agreement orclupdateschedule attribute.

```
ldapmodify -D "cn=orcladmin" -w administrator_password -h my_host -p port -f
mod.ldif
```

3. Restart the directory replication server.

Managing Replication

Once you have installed and configured replication, you can view or modify the default values for replication-related objects. This section contains these topics:

- Viewing and Modifying Directory Replication Server Configuration Parameters
- Viewing and Modifying Parameters for Particular Replica Nodes
- Modifying Parameters for Replication Agreements
- Changing the Replication Administrator's Password on All Nodes
- Managing the Change Log
- Modifying the Speed of Directory Replication

See Also:

- "Replication Agreements" on page 24-12
- "The Replica Subentry" on page 24-14

Note: No change to any configuration parameter or replication agreement takes effect until the replication server is restarted.

Viewing and Modifying Directory Replication Server Configuration Parameters

Table B–30 on page B-37 lists and describes the directory replication server configuration parameters. These 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.

Viewing Configuration Parameters of the Directory Replication Server by Using Oracle Directory Manager

To view configuration parameters of the directory replication server:

1. In the navigator pane, expand in succession **Oracle Internet Directory Servers**, ***directory server instance***, **Server Management**.
2. Select **Replication Server**. The following tab pages appear in the right pane.
   - **Active Replication Servers**, which tells you which directory replication servers are now running
   - **Replication Status**, which tells you the number of the last change applied from each supplier to each consumer in the DRG
   - **Changelog Subscriber Status**, which lists subscribers to the change log, and gives the number of the last change applied from this node

Modifying Configuration Parameters of the Directory Replication Server by Using Oracle Directory Manager

To modify configuration parameters of the directory replication server:

1. In the navigator pane, expand **Oracle Internet Directory Servers**, ***directory server instance***, **Server Management**, **Replication Server**.

2. Select the replication configuration set whose parameters you want to modify. The corresponding tab pages appear in the right pane.
3. In the **General** tab page, modify the fields as appropriate. Table C–14 on page C-13 describes the fields in this tab page.
4. For the Oracle9*i* Advanced Replication-based agreements, in the **ASR Agreement** tab page, modify the fields as appropriate. Table C–15 on page C-13 describes the fields in this tab page.
5. Restart the directory replication server to effect your changes.

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.

Modifying Directory Replication Server 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-31.

Table B–30 on page B-37 lists and described the replication server configuration parameters. As noted in that table, the modifiable replication configuration parameters are:

- `orclChangeRetryCount`
- `orclThreadsPerSupplier`

Example: 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 the example file `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 -D "cn=orcladmin" -w administrator_password -h my_host -p 389 -f
mod.ldif
```

3. Restart the directory replication server.

Example: 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 the example file `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 -D "cn=orcladmin" -w administrator_password -h my_host -p 389 -f
mod.ldif
```

3. Restart the directory replication server.

> **See Also:** "Restarting Oracle Internet Directory Server Instances" on page A-16 for instructions on restarting the directory replication server

Viewing and Modifying Parameters for Particular Replica Nodes

To modify a particular replica node, you modify the replica subentry. Table B–31 on page B-38 lists and describes the parameters you can modify in the replica subentry.

Note: Because the directory replication server reads replication node objects from the consumer, you must apply all changes to the consumer and, optionally, to the supplier.

> **See Also:** "The Replica Subentry" on page 24-14 for more information about the replica subentry

Viewing and Modifying Parameters for a Particular Replica Node by Using Oracle Directory Manager

To view and modify a particular replica node by using Oracle Directory Manager:

1. In the navigator pane, expand **Oracle Internet Directory Servers**, ***directory server instance***, **Replication Management**.
2. Select the replica node you want to view or modify. The corresponding tab pages appear in the right pane.
3. In the **General** tab page, you can modify the fields as appropriate. Table C–16 on page C-14 describes the fields in this tab page.
4. The **Replica Agreements** tab page enables you to view the details of the replication agreement in which the specified node participates. The columns in this tab page are described in Table C–17 on page C-15.
5. After you have viewed and modified the replica node, restart the directory replication server.

Modifying a Particular Replica Node 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-31.

Example: Modifying the orclReplicaURI Attribute for a Particular Replica Node The directory replication server uses the `orclReplciaURI` attribute value of the replica subentry to locate the directory server for that replica. If the port or host where the directory server is running is changed, then this attribute must be modified accordingly.

1. Edit the example file `mod.ldif` as follows:

```
Dn: orclreplicaid=unique_replica_identifier, cn=replication configuration
Changetype:modify
Replace:orclReplicaURI
OrclReplicaURI: ldap://host_name:port_number/
```

2. Use ldapmodify to update the replica subentry `orclreplicauri` attribute.

```
ldapmodify -D "cn=orcladmin" -w administrator_password -h my_host -p 389 -f
mod.ldif
```

3. Restart the directory replication server.

Example: Modifying the orclReplicaSecondaryURI Attribute for a Particular Replica The directory replication server uses the `orclReplicaSecondaryURI` attribute value

as an alternate location to contact the directory server for a particular replica. A user can add an alternate `ldapURI` attribute at which the directory server can be contacted for that particular replica. To add additional `ldapURI` attribute:

1. Edit the example file `mod.ldif` as follows:

```
Dn: orclreplicaid=unique_replica_identifier, cn=replication configuration
Changetype:modify
add:orclReplicaSecondaryURI
OrclReplicaSecondaryURI: ldap://host_name:port_number/
```

2. Use ldapmodify to update the replica subentry OrclReplicaSecondaryURI attribute.

```
Ldapmodify -D "cn=orcladmin" -w administrator_password -h my_host -p 389 -f
mod.ldif
```

3. Restart the directory replication server.

Example: Modifying the orclReplicaState Attribute for a Particular Replica

`OrclReplicaState` represents the state of a particular replica. To bootstrap (re-initialize) a replica, update this attribute in the following manner:

1. Edit the example file `mod.ldif` as follows:

```
Dn: orclreplicaid=<unique replica identifier>, cn=replication configuration
Changetype:modify
replace:orclReplicaState
OrclReplicaState: 0
```

2. Use ldapmodify to update the replica subentry `orclreplicastate` attribute.

```
Ldapmodify -D "cn=orcladmin" -w administrator_password -h my_host -p 389 -f
mod.ldif
```

3. Restart the directory replication server.

Modifying Parameters for Replication Agreements

This section contains instruction for modifying replication agreements that are based on both Oracle9*i* Advanced Replication and LDAP.

Modifying Parameters for Replication Agreements Based on Oracle9*i* Advanced Replication

Replication agreement parameters based on Oracle9*i* Advanced Replication are stored in replication agreement entries, which have the following DN:

```
orclAgreementID=000001,cn=replication configuration
```

Note:

- For replication agreements based on Oracle9*i* Advanced Replication, in the parameter `DirectoryReplicationGroupDSAs`, enter the host names for all of the nodes in the DRG. This list must be identical on all the nodes.
- For Oracle Internet Directory 10*g* (9.0.4), only one replication agreement based on Oracle9*i* Advanced Replication can be used. The DN of this replication agreement is `orclagreementid=000001,cn=replication configuration`.
- 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 Agreements Based on Oracle9i Advanced Replication by Using Oracle Directory Manager" on page 25-42
- "Managing Replication Agreements Based on Oracle9i Advanced Replication by Using ldapmodify" on page 25-42

Viewing and Modifying Replication Agreements Based on Oracle9*i* Advanced Replication by Using Oracle Directory Manager To view and modify replication agreement parameters by using Oracle Directory Manager:

1. In the navigator pane, expand in succession Oracle Internet Directory **Servers**, ***directory server instance***, **Replication Management**. The following tab pages appear in the right pane:
   - **Replication Status**, which tells you the number of the last change applied from each supplier to each consumer in the DRG
   - **Replica Status**, which tells you the state of the replica—that is, whether it is online, offline, or in the bootstrapping process.
   - **Changelog Subscriber**, which lists subscribers to the change log, and gives the number of the last change applied from this node
   - **ASR Agreement**, in which you can view and modify the information for a Oracle9*i*Advanced Replication-based replication agreement. The fields in this tab page are described in Table C–15 on page C-13.

   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.

2. 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**.

Managing Replication Agreements Based on Oracle9*i* Advanced Replication by Using ldapmodify Table B–32 on page B-38 lists and describes the replication agreement parameters and indicates those that you can modify.

To add more nodes to the values in a replication agreement entry, run ldapmodify at the command line, referencing an LDIF-formatted file.

Example 1: Adding Nodes to a Replication Agreement

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=replication configuration
changetype: modify
```

```
add: orcldirreplgroupdsas
orcldirreplgroupdsas: hollis
orcldirreplgroupdsas: eastsun-11
```

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

```
ldapmodify -D "cn=orcladmin" -w administrator_password -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=replication configuration`. 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 previous sample LDIF file—in the `orclDirReplGroupDSAs` parameter on each node in the replicated environment before you start the replication process.

"Adding a Node to a Multimaster Replication Group" on page 25-13 explains the process of adding a new node to a replication environment.

Because Oracle Internet Directory 10*g* (9.0.4) supports only one configuration set for the directory replication server, you do not need to specify a configuration set.

Example 2: Modifying the orclExcludedNamingcontexts Attribute for an Oracle9*i* Advanced Replication Replica Agreement

In a replication agreement based on Oracle9*i* Advanced Replication, the directory replication server uses the value of the `orclExcludedNamingcontexts` attribute of the replica agreement entry to specify the top level subtrees to be excluded from replication.

In this example, two top level naming contexts—`c=us` and `c=uk`—are excluded from Oracle9*i* Advanced Replication.

1. Edit the example file `mod.ldif` as follows:

```
dn: orclAgreementID=000001, cn=replication configuration
Changetype:modify
Replace: orclExcludedNamingcontexts
```

```
orclExcludedNamingcontexts: c=us
orclExcludedNamingcontexts: c=uk
```

2. Use ldapmodify to update the replication agreement `orclupdateschedule` attribute.

```
ldapmodify -D "cn=orcladmin" -w administrator_password -h consumer_host -p
port -f mod.ldif
```

3. Restart the directory replication server.

Modifying Parameters for Replication Agreements Based on LDAP

LDAP-based replication agreement parameters are stored in replication agreement entries, which have the following DN:

```
orclAgreementID=id number,orclReplicaId=replica id, cn=replication configuration
```

> **Note:** Ensure that the agreement is identical at both the supplier and the consumer. The last applied change number and the naming context are read from the agreement at the supplier node. The other agreement attributes are read from the consumer.

Viewing and Modifying LDAP-Based 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 in succession **Oracle Internet Directory Servers**, ***directory server instance***, **Replication Management**, **Replica Node:** ***replica identifier***.
2. Select the replica agreement you want to view or modify. The following tab pages appear in the right pane:
   - **General**, in which you can view and modify LDAP based replication agreement information. The fields in this tab page are described in Table C–17 on page C-15.
   - **Replica Naming Context**, in which you can view, add, delete, and modify LDAP naming context 0bjects. The fields in this tab page are described in Table C–18 on page C-16.

Modifying LDAP-Based Replication Agreement Parameters by Using ldapmodify Table B–32 on page B-38lists and describes the replication agreement parameters and indicates those that you can modify.

Example 1: Modifying the orclUpdateSchedule Attribute for a Particular Replica Agreement

The directory replication server uses the `orclupdateschedule` attribute value of the replica agreement entry as time interval in minutes to determine how often the replication server process the new change logs from the supplier.

This example shows that replication server will process new change logs from the supplier for every minute.

1. Edit the example file mod.ldif as follows:

```
dn: orclAgreementID=id_number,orclReplicaId=replica_identifier,
cn=replication configuration
Changetype:modify
Replace: orclupdateschedule
orclupdateschedule: 1
```

2. Use ldapmodify to update the replication agreement orclupdateschedule attribute.

```
ldapmodify -D "cn=orcladmin" -w administrator_password -h consumer_host -p
port -f mod.ldif
```

3. Restart the directory replication server.

Example 2: Modifying the orclLastAppliedChangeNumber Attribute for Particular Replica Agreement

The directory replication server uses the value `orclLastAppliedChangeNumber` attribute to determine the number of last applied change log processed by the consumer.

The modification of the `orclLastAppliedChangeNumber` attribute must be applied against the supplier node since replication server reads `orclLastAppliedChangeNumber` from the same duplicate agreement at the supplier.

In this example, orclLastAppliedChangeNumber attribute of the duplication agreement at the supplier is set to 700, which indicates all change logs with changelog number prior to 700 has been processed by the replication server.

Note: Do not modify the `orclLastAppliedChangeNumber` attribute except as instructed during partial replication add node procedure.

1. Edit the example file `mod.ldif` as follows:

```
dn: orclAgreementID=id_number,orclReplicaId=replica_id,cn=replication
configuration
Changetype:modify
Replace: orclLastAppliedChangeNumber
orclLastAppliedChangeNumber: 700
```

2. Use ldapmodify to update the replication agreement `orclupdateschedule` attribute at the supplier.

```
ldapmodify -D "cn=orcladmin" -w administrator_password -h supplier_host -p
port -f mod.ldif
```

3. Restart the directory replication server.

Changing the Replication Administrator's Password on All Nodes

You can change the password for the replication administrator database account on all nodes of a DRG by using the `-chgpwd` utility of the Replication Environment Management Tool. To launch this utility, enter:

```
remtool -chgpwd
```

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 you must run the Replication Environment Management Tool again.

Also, you can change the password of the replication DN of a replica by using the `-pchgpwd` utility. To launch this utility, enter:

```
remtool -pchgpwd
```

See Also: "The Replication Environment Management Tool" on page A-62 for more information about using this tool

Managing the Change Log

Oracle Directory Manager enables you to view the last 25 changes you performed, listing them by change log number, the type of operation—namely, add, modify, or delete—in which each occurred, and the entry on which each was made. It allows you select a particular change to see more specific details about it.

To manage the change log, in the navigator pane expand in succession **Oracle Internet Directory Servers**, ***directory server instance***, then select **Change Log Management**. The right pane lists the last 25 changes, beginning with the most recent. It tells you the change number, the type of operation in which each change occurred, and the entry on which the change was made.

To see the details of a particular change, in the right pane, select the change, then choose **View Properties**. The Change Log window appears. The fields for the Change Log window are listed and described in Table C–19 on page C-17.

Modifying the Speed of Directory Replication

In the default configuration for replication, the `orclupdateschedule` attribute is set to a value of `1`, representing 1 minute. You can shorten the replication processing time by changing the value of the `orclupdateschedule` attribute to `0`, representing 1 second.

Modifying the Speed of Directory Replication When Using Oracle9*i* Advanced Replication

In directory replication based on Oracle9*i* Advanced Replication, the default configuration achieves a processing time that is approximately 2.5 minutes:

- 1 minute for the supplier to prepare the change for sending to the consumer
- 30 seconds for Oracle9*i* Advanced Replication to push the change to the consumer
- 1 minute for the consumer to apply the change

In the case of Oracle9*i* Advanced Replication, changing the default value for the `orclupdateschedule` attribute to `0` results in a replication time of 32 seconds. To do this:

1. Edit `mod.ldif` as follows:

```
dn: orclagreementid=orclagreementid=000001, cn=replication configuration,
cn=replication configuration
changetype:modify
```

```
replace: orclupdateschedule
orclupdateschedule: 0
```

2. Upload `mod.ldif` as follows:

```
ldapmodify -h host name -p port number -v -f mod.ldif
```

3. Restart the directory replication server

```
oidctl connect=connect string server=oidrepld instance=instance number
restart
```

Modifying the Speed of Directory Replication When Using LDAP-Based Replication

In LDAP-based directory replication, the default configuration achieves a processing time that is approximately 1 minute during which the change is retrieved from the supplier and applied to the consumer. Changing the default value for the `orclupdateschedule` attribute to `0` results in a replication time of 1 second. To do this:

1. Edit `mod.ldif` as follows:

```
dn: orclagreementid=agreement ID,orclreplicaid=replica ID,
cn=replication configuration
changetype:modify
replace: orclupdateschedule
orclupdateschedule: 0
```

2. Upload `mod.ldif` as follows:

```
ldapmodify -h host name -p port number -v -f mod.ldif
```

3. Restart the directory replication server

```
oidctl connect=connect string server=oidrepld instance=instance number
restart
```

Example: Installing and Configuring a Multimaster Replication Group with Fan-Out

To help you install and configure a multimaster replication group with fan-out, this section offers an example with three systems as described in Table 25–2.

Table 25–2 Nodes in Example of Partial Replication Deployment

Node	Host Name	Port
Node1	mycompany1.com	3000
Node2	mycompany2.com	4000
Node3	mycompany3.com	5000

In this example, the user requirements are:

- Node1 and Node2 must be synchronized so that changes made on either node are replicated to the other— but the naming context `cn=private users, cn=mycompany` is to be excluded from this replication.
- The naming context `ou=Americas,cn=mycompany` on node3 is to be partially synchronized from Node2 so that only changes made under `ou=Americas, cn=mycompany` on Node2 are replicated to Node3. The following are to be excluded from this replication:
  - Changes made under `cn=customer profile, ou=Americas, cn=mycompany`
  - Changes in the attribute `userpassword`.

To meet the first requirement in this example, we set up a multimaster replication group for Node1 and Node2. To meet the second, we set up a partial replica for Node2 and Node3.

This section contains these topics:

- Task 7: Start the Replication Servers on All Nodes in the DRG

Task 1: Set up the Multimaster Replication Group for Node1 and Node2

To set up the multimaster replication group for Node1 and Node2, follow Tasks 1 through 5 in the section "Installing and Configuring a Multimaster Replication Group" on page 25-2.

Task 2: Configure the Replication Agreement

In the replication agreement between Node1 and Node2, specify the value for the `orclExcludedNamingcontexts` attribute as `cn=private users,cn=mycompany`. To do this:

1. Edit the example file `mod.ldif` as follows:

```
dn: orclAgreementID=000001,cn=replication configuration
Changetype:modify
Replace: orclExcludedNamingcontexts
orclExcludedNamingcontexts: cn=private users,cn=mycompany
```

2. Use ldapmodify to update the replication agreement `orclExcludedNamingcontexts` attribute at both Node1 and Node2. To do this, enter:

```
ldapmodify -D "cn=orcladmin" -w administrator_password -h mycompany1.com -p
3000 -f mod.ldif
ldapmodify -D "cn=orcladmin" -w administrator_password -h mycompany2.com -p
4000 -f mod.ldif
```

Task 3: Start the Replication Servers on Node1 and Node2

To do this, follow the instructions in "Task 6: Start the Replication Servers on All Nodes in the DRG" on page 25-12.

Task 4: Test the Directory Replication

To do this, follow the instructions in "Task 7: Test Directory Replication" on page 25-13.

Task 5: Install and Configure Node3 as a Partial Replica of Node2

If you want to use the bootstrap capability of partial replication, then follow Tasks 1 through 3 in "Configuring an LDAP-Based Replica by Using Automatic Bootstrapping" on page 25-24.

If you want to configure the replica by using the ldifwrite tool, then follow Tasks 1 through 7 in "Configuring an LDAP-Based Replica by Using the ldifwrite Tool" on page 25-26.

Identify Node2 as the supplier and Node3 as the consumer.

Task 6: Customize the Partial Replication Agreement

To do this:

1. Start the directory server at the consumer, namely, Node3.
2. To achieve the second requirement in this example, we need to configure the default replication parameters of the partial replica between Node2 and Node3. In partial replication, the `cn=oraclecontext` naming context is replicated by default. You can choose not to replicate it by deleting it at both the supplier and the consumer.

```
ldapdelete -D "cn=orcladmin" -w administrator_password -h mycompany2.com -p
4000
"cn=includednamingcontext000001,cn=replication
namecontext,orclagreementid=000002,orclreplicaid==<node2_replica_
id>,cn=replication configuration"

ldapdelete -D "cn=orcladmin" -w administrator_password -h mycompany3.com -p
5000
"cn=includednamingcontext000001,cn=replication
namecontext,orclagreementid=000002,orclreplicaid==<node2_replica_
id>,cn=replication configuration"
```

To replicate the naming context `ou=Americas,cn=mycompany`, and to exclude from replication the naming context `cn=customer profile, ou=Americas, cn=mycompany` and the attribute `userpassword`, create a naming context object as follows:

1. Edit the example file `mod.ldif` as follows:

```
dn: cn=includednamingcontext000002,cn=replication
namecontext,orclagreementid=000002,orclreplicaid=node2_replica_
id,cn=replication configuration
orclincludednamingcontexts: ou=Americas,cn=mycompany
orclexcludednamingcontexts: cn=customer profile, ou=Americas, cn=mycompany
orclexcludedattributes: userpassword
objectclass: top
objectclass: orclreplnamectxconfig
```

2. Use ldapadd to add the partial replication naming context object at both Node2 and Node3.

```
ldapadd -D "cn=orcladmin" -w administrator_password -h mycompany2.com -p
4000 -f mod.ldif
ldapadd -D "cn=orcladmin" -w administrator_password -h mycompany3.com -p
5000 -f mod.ldif
```

If you decide to use the automatic bootstrap capability of partial replication, then do the following:

1. Edit the example file mod.ldif as follows:

```
dn: orclreplicaid=<node2's replica id>,cn=replication configuration
changtype: modify
replace: orclreplicastate
orclreplicastate: 0
```

2. Use ldapmodify to modify the partial replica `orclreplicastate` attribute at both Node2 and Node3.

```
ldapmodify -D "cn=orcladmin" -w administrator_password -h mycompany2.com -p
4000 -f mod.ldif
ldapmodify -D "cn=orcladmin" -w administrator_password -h mycompany3.com -p
5000 -f mod.ldif
```

Task 7: Start the Replication Servers on All Nodes in the DRG

To do this, follow the instructions in "Task 9: Start the Directory Replication Server on the Consumer Replica" on page 25-30.

26

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. It contains these topics:

- About High Availability and Failover for Oracle Internet Directory
- Oracle Internet Directory and the Oracle Technology Stack
- Failover Options on Clients
- Failover Options in the Public Network Infrastructure
- High Availability and Failover Capabilities in Oracle Internet Directory
- Failover Options in the Private Network Infrastructure
- High Availability Deployment Examples

> **See Also:** "Directory Replication and High Availability" for information about high availability and failover in clustered environments

About High Availability and Failover for Oracle Internet Directory

Oracle Internet Directory provides the high degree of system availability that mission-critical applications require. It does this by enabling:

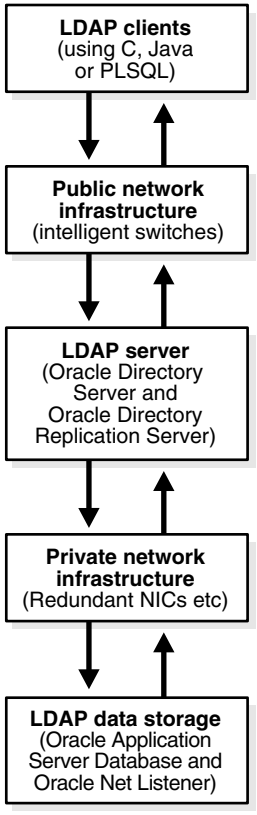
- All components in the system to facilitate redundancy
- All interfaces to facilitate failure recognition and recovery, called **failover**
- Integration of application-independent network failover capabilities in the overall deployment

Oracle products are commonly targeted for high availability environments and hence necessary capabilities are built into all layers of the Oracle technology stack. Typically, it is not necessary to employ every failover capability in every component.

Oracle Internet Directory and the Oracle Technology Stack

Figure 26–1 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 26–1 Oracle Internet Directory/Oracle 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 these layers.

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 points 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.

Clients should cache the information in the alternate server list for use in the event that the primary server becomes unavailable.

Setting the Alternate Server List by Using Oracle Directory Manager

To set the alternate server list:

1. In the navigator pane, expand Oracle Internet Directory Servers, then select a server instance. System operational attributes appear in the right pane.
2. In the Alternate Server field, enter the name or names of alternate servers.
3. Choose OK.

See Also:

- RFC 2251 at `http://www.ietf.org` for details about the usage of `altServer` attribute
- "Managing Attributes by Using Command-Line Tools" on page 6-17 for instructions about setting the `AltServer` attribute

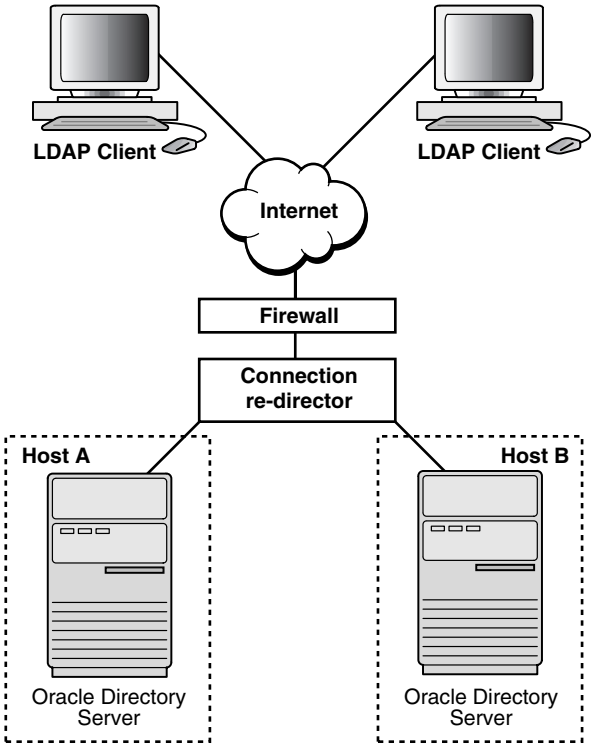
Failover Options in the Public Network Infrastructure

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 26–2 illustrates a typical Internet deployment of Oracle Internet Directory with network-level failover enabled.

Figure 26–2 Network-Level Failover



In Figure 26–2, the Oracle directory servers (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.

High 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.

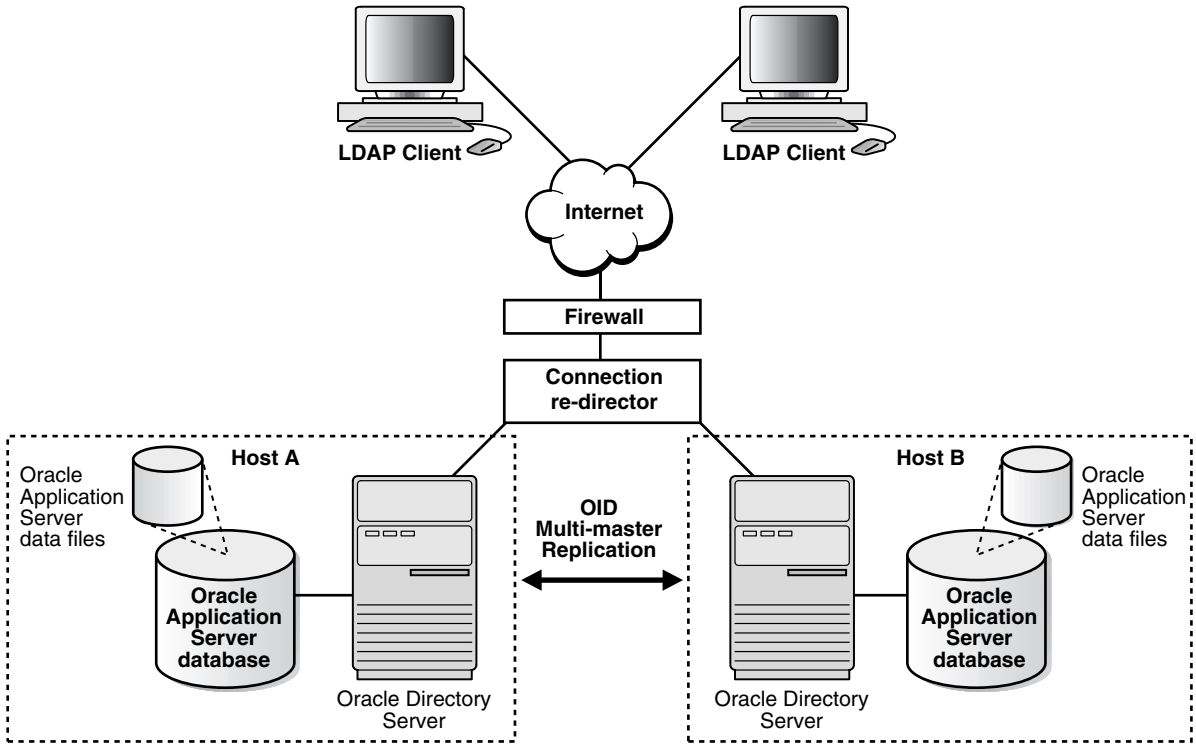
Redundant Links

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

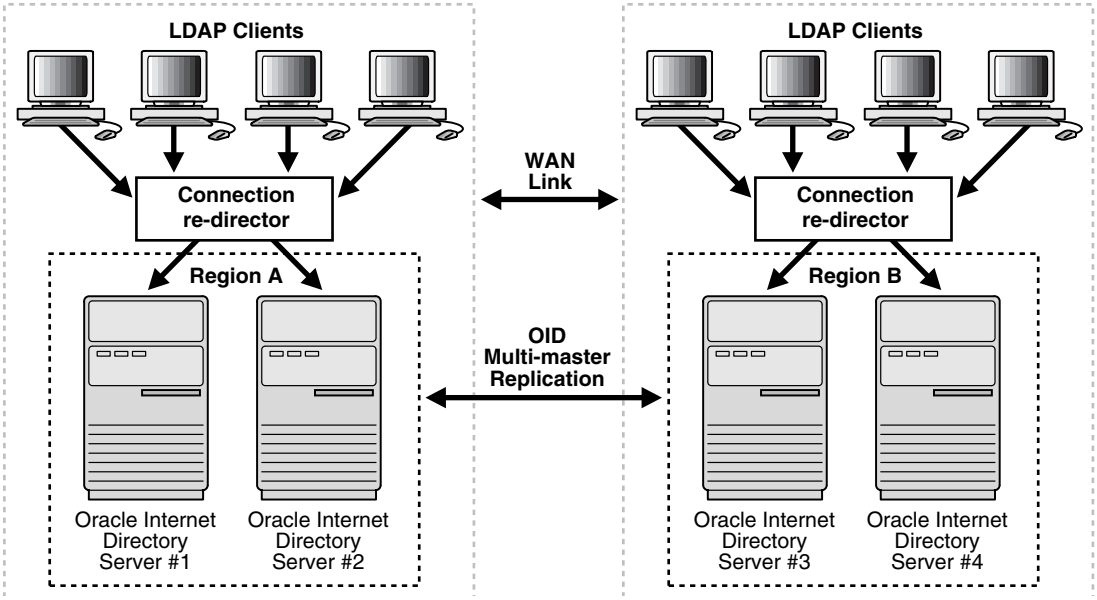
In Figure 26–3, the database and Oracle directory server (LDAP server) are co-resident on the same computer. Changes made on one directory server instance are reflected on the second directory server instance through multimaster replication. When a failure of the directory server or database server on 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 26–3 Deployment Example (Two Oracle Internet Directory Nodes in Replication)



As Figure 26–4 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 could potentially represent a continent or a country.

Figure 26–4 Deployment Example 2



27

Rack-Mounted Directory Server Configurations

This chapter describes rack-mounted directory server configuration, which provides high availability of a directory server. This configuration involves running multiple directory server instances on different hardware nodes. The directory servers are connected to the same directory store, which is an Oracle9*i* Database Server.

This chapter contains these topics:

Note: This chapter describes a high availability configuration for directory servers. It does not address high availability for database servers that store directory data. For the latter, use one of the standard high availability configurations for database servers such as Oracle Real Applications Clusters or Data Guard.

See Also: Chapter 29, "The Directory in an Oracle9i Real Application Clusters Environment"

About Rack-Mounted Directory Server Configurations

In a rack-mounted directory server configuration, multiple directory server instances run on different hardware nodes but connect to the same directory store, which is an Oracle9*i* Database Server.

The key benefits of the rack-mounted configuration are:

- Scalability and Performance

  Load balancing is achieved by redirecting LDAP clients to multiple directory nodes. Each additional hardware node added to the directory node increases both the number of concurrent clients that can be supported and the throughput of LDAP operations.

- High Availability of Directory Servers

  High availability of directory servers can be achieved through a network re-director that changes the direction of the LDAP request on the failed directory server node to the other ones that are still running.

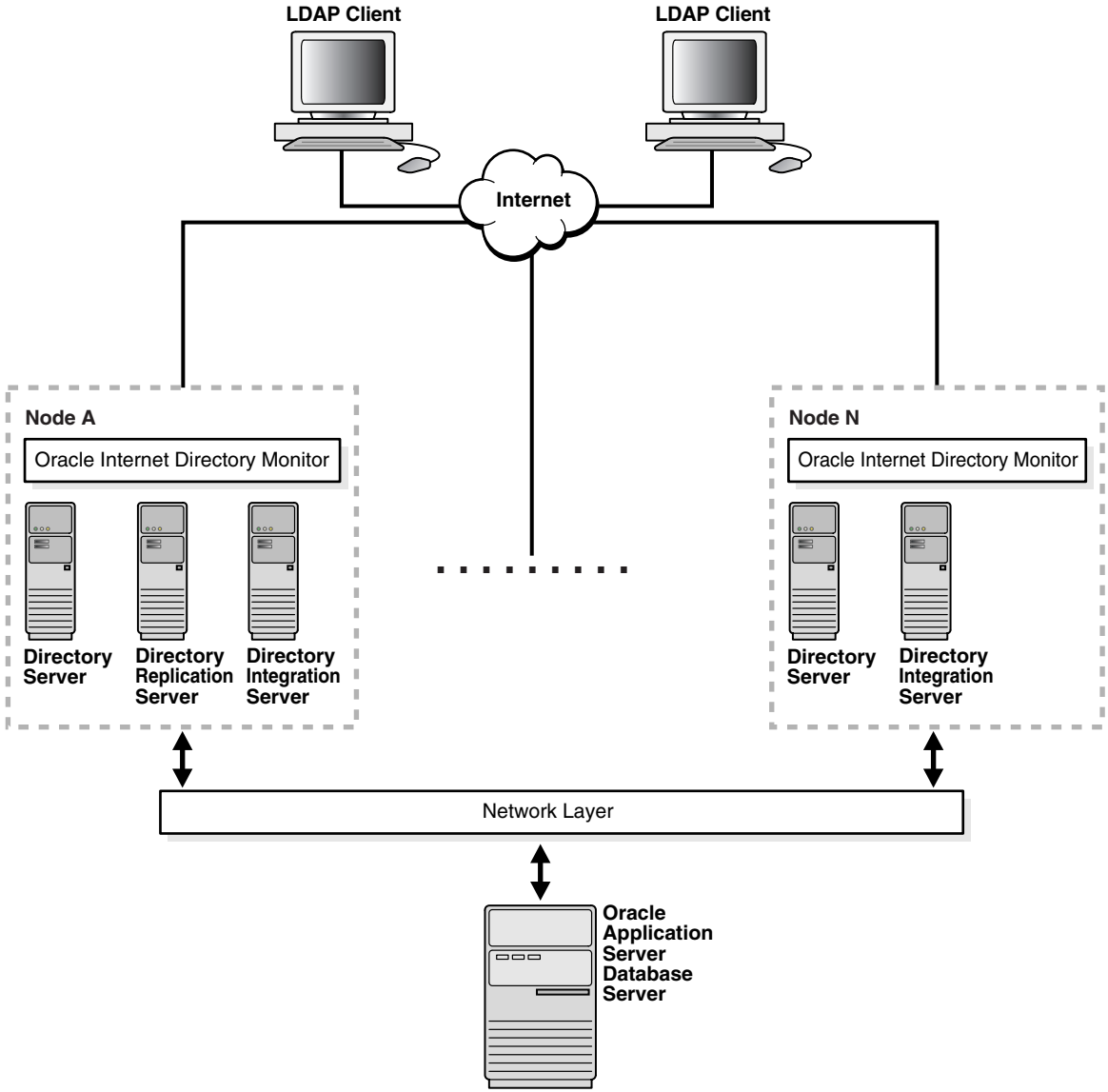
- Reduced cost of ownership

  A rack-mounted directory server configuration requires only low-cost hardware to achieve all the benefits of high performance, high availability, and scalability.

Architecture of the Rack-Mounted Directory Server Configuration

Figure 27–1 on page 27-3 shows the architecture of a rack-mounted directory server configuration.

Figure 27–1 Architecture of a Rack-Mounted Directory Server Configuration



As Figure 27–1shows, in a rack-mounted environment, a replication server can reside on one node only. If, after 10 tries, the OID Monitor on one node fails to start

either a directory replication server or a directory integration server, then it pushes the start request to the OID Monitor on another node.

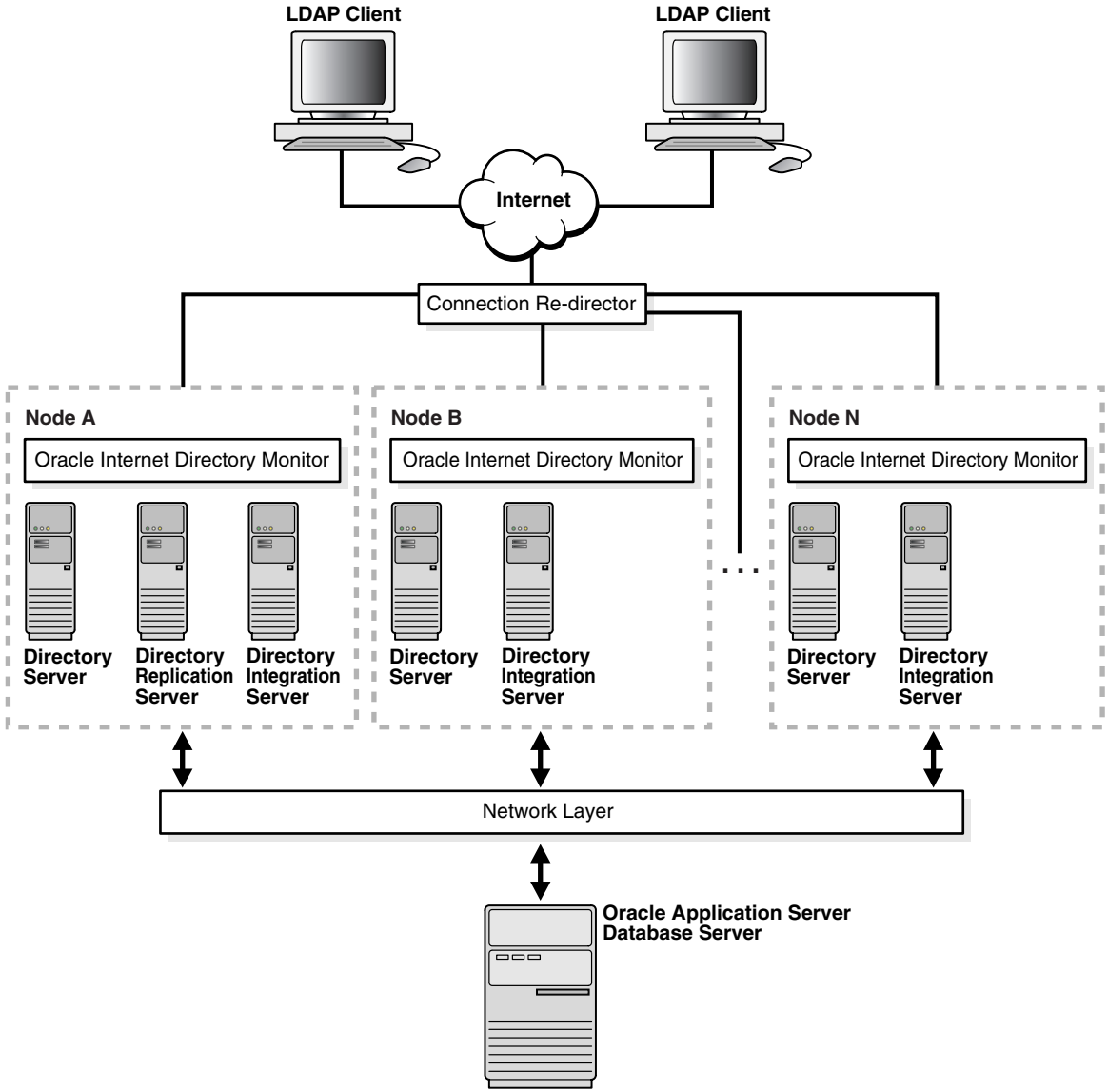
Multiple instances of the Oracle directory integration and provisioning server should not be started using the same configuration set entry.

Load Balancing for High Availability

Load balancing needed for high availability of directory servers can be achieved through a network re-director that changes the direction of the LDAP request on the failed directory server node to the other nodes that are still running.

Figure 27–2 on page 27-5 shows load balancing in a rack-mounted directory server configuration.

Figure 27–2 Load Balancing in a Rack-Mounted Directory Server Configuration

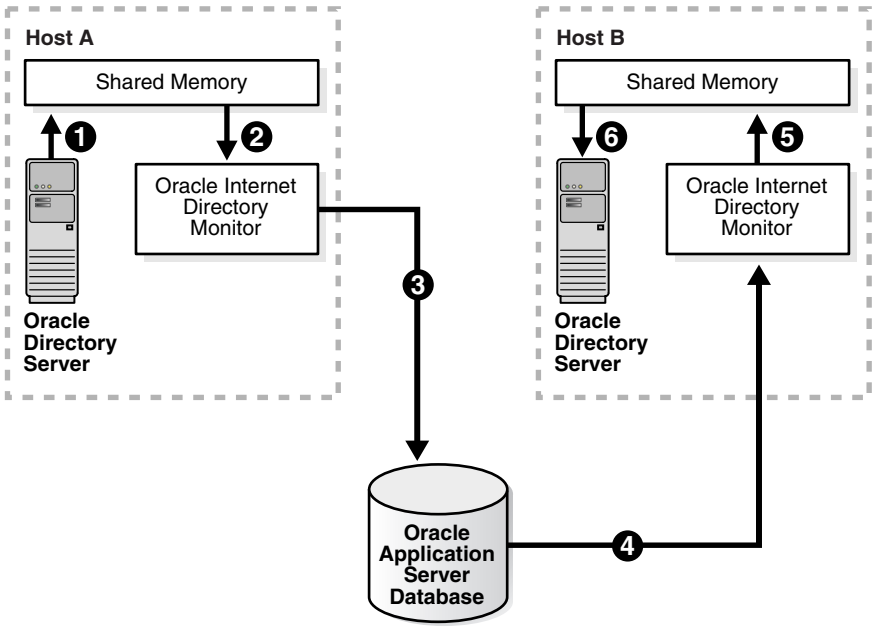


As Figure 27–2 shows, when LDAP clients seek to connect to a directory, a connection re-director handles that connection. If a directory server node has failed, then this re-director connects the client to one that is running.

Metadata Synchronization in a Rack-Mounted Directory Server Environments

In a rack-mounted directory server environment, it is necessary to synchronize the metadata—for example, definitions of object classes, attributes, matching rules, ACPs, and password policies—on all the directory server nodes. Figure 27–3 and the accompanying text exemplify the process in which directory server metadata is synchronized between two directory server nodes, Host A and Host B, in a rack-mounted environment.

Figure 27–3 Metadata Synchronization Process in Rack-Mounted Environments



In the example in Figure 27–3, metadata in a rack-mounted environment is synchronized as follows:

1. On Host A, the directory server writes metadata changes to the shared memory on that same host.
2. OID Monitor on Host A polls the shared memory on that same host. When it discovers a change in the metadata, it retrieves the change.
3. OID Monitor sends the change to the Oracle9*i* Database Server, which is the repository for the directory server metadata in the rack-mounted environment.
4. OID Monitor on Host B polls Oracle9*i* Database Server for changes in directory server metadata, and retrieves those changes.
5. OID Monitor on Host B sends the change to the shared memory on that same host.
6. The directory server on Host B polls the shared memory on that same host for metadata changes. It then retrieves and applies those changes.

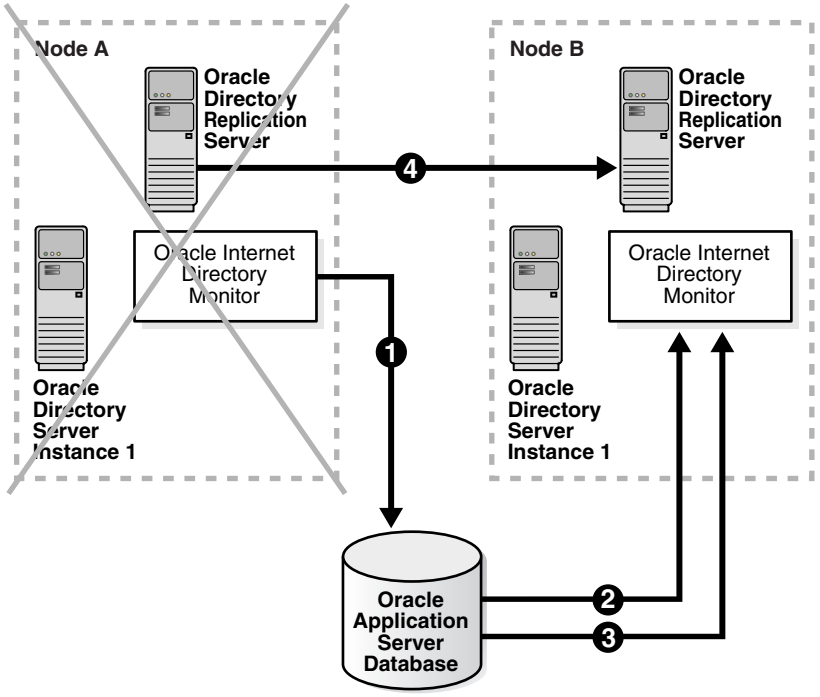
How Failover Works in a Rack-Mounted Directory Server Environment

In a rack-mounted environment, the OID Monitor on each node reports to the other nodes that it is running by sending a message to the Oracle9*i* Database Server every 10 seconds. When it does this, it also polls the database server to verify that all other directory server nodes are also running. If, after 250 seconds, an OID Monitor on one of the nodes has not reported that it is running, then the other directory server nodes regard it as having failed. At this point, the following occurs on one of the other nodes that are still running:

1. The OID Monitor on that node brings up the processes that were running on the failed node.
2. The directory server on that node continues processing the operations that were previously underway on the failed node.
3. The OID Monitor on that node logs that it has brought up the processes that were previously running on the failed node.

Figure 27–4 on page 27-8 and the accompanying text exemplify this process on two hypothetical nodes, Node A and Node B.

Figure 27–4 Example of Failover in a Rack-Mounted Environment



As the example in Figure 27–4 shows, the failover process in a rack-mounted environment follows this process:

1. Every 10 seconds, the OID Monitor on Node A reports that it is running by sending a message to the database.
2. The OID Monitor on Node B polls the database to learn which, if any, of the other nodes may have failed.
3. When OID Monitor on Node B learns that Node A has not responded for 250 seconds, it regards Node A as having failed. It then retrieves from the database the necessary information about the Oracle Internet Directory servers that were running on Node A. In this example, it learns that the directory replication server had been running on Node A.
4. Because a directory replication server was not already running on Node B, the OID Monitor on Node B starts a directory replication server that corresponds to the directory replication server previously running on Node A.

See Also: "Oracle Internet Directory Architecture" on page 2-14 for information about directory server nodes, directory server instances, and the kinds of directory metadata stored in the database

Rules for Managing a Rack-Mounted Directory Server Environment

In a rack-mounted directory server environment, the port number for the directory server must be the same on all the nodes.

The time value on all nodes should be synchronized using Greenwich mean time so that there is a discrepancy of no more than 250 seconds between them.

When you restart a failed node, you must manually start all of the servers previously running on that node.

If you change the password to the Oracle Internet Directory-designated database, then you must update each of the other nodes in the rack-mounted environment.

See Also:

- "OID Database Password Utility (oidpasswd) Syntax" on page A-131 for instructions on how to change the password to the Oracle Internet Directory-designated database
- Starting and Stopping Oracle Internet Directory Servers on Either a Virtual Host or a Rack Node on page A-17

Installation of a Rack-Mounted Directory Server

To install a rack-mounted directory server:

1. On the database node, install Oracle Internet Directory.

   When prompted to select a product, choose the Oracle Application Server Infrastructure.

   When prompted to choose one of various installation types, choose the Oracle Application Server Metadata Repository installation type.

2. On the first rack-mounted node, install OracleAS Portal Infrastructure > Identity Management.

   When prompted for configuration options, select Oracle Internet Directory.

When prompted to specify repository, enter `sys dba username/password`, the name of the database node, and the listener port number. This installs Oracle Internet Directory and configures the base schema against the remote database.

3. On each other node:

   a. Install OracleAS Portal infrastructure > Identity Management.

   b. When prompted to select configuration options, deselect all configuration options including Oracle Internet Directory. This installs Oracle Internet Directory but does not attempt to configure the schema against the remote database.

   c. Copy the `tnsnames.ora` from the first rack-mounted node into the *`ORACLE_HOME`*`/network/admin` directory.

   d. Execute the OID Database Password Utility to set up the wallet. To do this, enter:

      ```
      oidpasswd connect=connect_string_for_the_database_node create_
      wallet=TRUE current_password=Oracle Application Server_admin_password.
      ```

      The *`connect_string_for_the_database_node`* is the same as that in the `tnsnames.ora` file you just copied from the first rack-mounted node in Step c.

      The *`Oracle Application Server_admin_password`* is the same as the one you used when installing on the first rack-mounted node.

   e. Start the directory server as follows.

      ```
      oidmon [connect=connect_string_for_the_database_node]
      [host=virtual/host_name][sleep=seconds] start

      oidctl connect=connect_string_for_the_database_node server=oidldapd
      instance=server_instance_number [configset=configset_number]
      [host=virtual/host_name][flags=' -p port_number -work maximum_number_of_
      worker_threads_per_server -debug debug_level -l change_logging' -server
      number_of_server_processes] start
      ```

      See Also:

      - "Starting the OID Monitor" on page A-4
      - "Starting an Oracle Directory Server Instance" on page A-7

28

Cold Failover Cluster Configuration

This chapter explains the cold failover cluster configuration, one of the high availability configurations for Oracle Internet Directory.

This chapter contains these topics:

About the Cold Failover Cluster Configuration

A **cluster** is a collection of interconnected usable whole computers that is used as a single computing resource. Hardware clusters provide high availability and scalability.

During **failover**, an application running on one cluster node is transparently migrated to another cluster node. During this migration, clients accessing the service on the cluster see a momentary outage and may need to reconnect once the failover is complete.

The cluster node on which the application runs at any given time is called the **primary node**. The cluster node to which the application is moved during a failover is called the secondary node.

In a hardware cluster, each physical node has its own physical IP address and physical host name. To present a single system image to the outside world, the cluster uses a dynamic IP address that can be moved to any physical node in the cluster. This is called the **virtual IP address**. The host name corresponding to this virtual IP address is called the logical or **virtual host name**. All network clients accessing a service on the cluster in a cold failover configuration use the virtual host name.

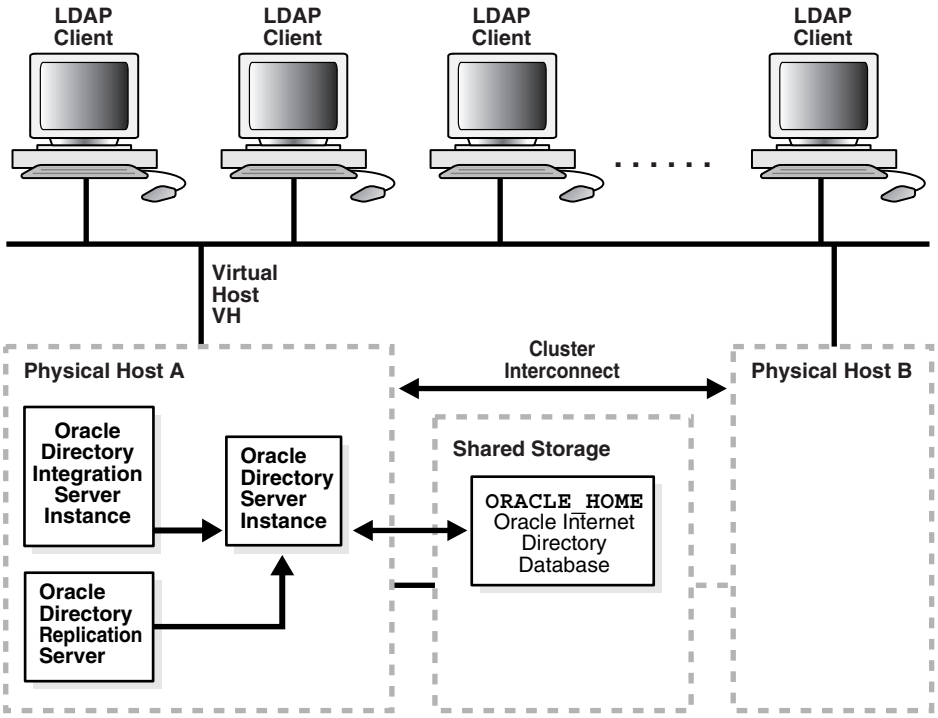
A **logical host** consists of one or more disk groups, and pairs of host names and IP addresses. It is mapped to a physical host in the cluster. This physical host impersonates the host name and IP address of the logical host.

Although each node is a usable whole computer, in most cases the storage subsystem is shared by all the nodes. In a cold failover cluster configuration, the shared storage subsystem hosts the Oracle Internet Directory installation—that is, the *ORACLE_HOME*—and at any given point in time is accessible by one active node.

The Simple Cold Failover Configuration

Figure 28–1 shows a simple cold failover configuration in which an unspecified number of LDAP clients connect to Physical Host A and Physical Host B.

Figure 28–1 Simple Cold Failover Configuration



In Figure 28–1 the primary cluster node is Physical Host A and the secondary cluster node is Physical Host B. There is only one software and database installation. Physical hosts A and B have access to the shared disk on which the Oracle Internet Directory software and database reside.

Physical host A is configured to host the virtual host VH and the installation on the virtual host VH. The Oracle Internet Directory processes are started on the virtual host VH. All LDAP clients talk to Oracle Internet Directory by using the virtual host name VH.

How to Ensure that Oracle Internet Directory Runs on the Virtual Host

You can start Oracle Internet Directory servers on virtual hosts by using either OID Monitor (oidmon) and OID Control Utility (oidctl), or by using the Oracle Directory Integration and Provisioning Server Registration Tool (`odisrvreg`).

When using the Oracle Directory Integration and Provisioning Server Registration Tool (`odisrvreg`), use the `lhost` parameter to specify the virtual host name.

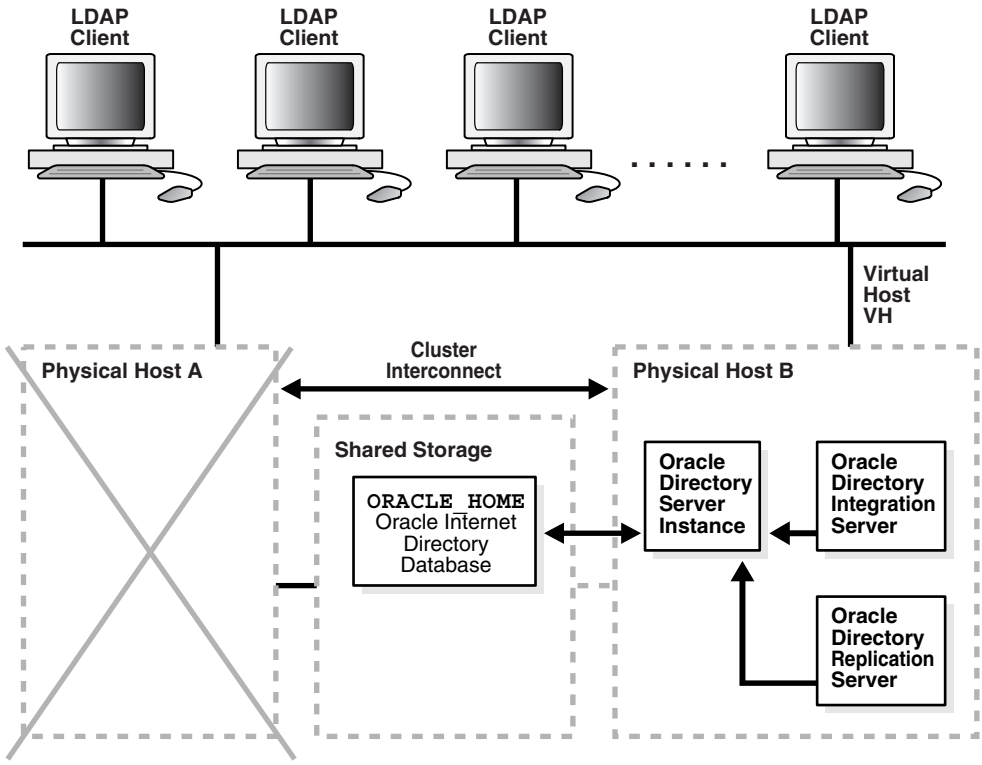
See Also:

- "Starting and Stopping Oracle Internet Directory Servers on Either a Virtual Host or a Rack Node" on page A-17
- "The Oracle Directory Integration and Provisioning Server Registration Tool (odisrvreg)" on page A-126

The Simple Cold Failover Process

To illustrate the cold failover process, Figure 28–2 shows the same environment as that in Figure 28–1 on page 28-3, but with Physical Host A having failed.

Figure 28–2 The Cold Failover Process



As Figure 28–2 shows, when Physical Host A fails or is shut down for maintenance purposes, the virtual host VH is migrated to Physical Host B. After the failover, you must restart the Oracle9*i* Database Server, the listener, and Oracle Internet Directory.

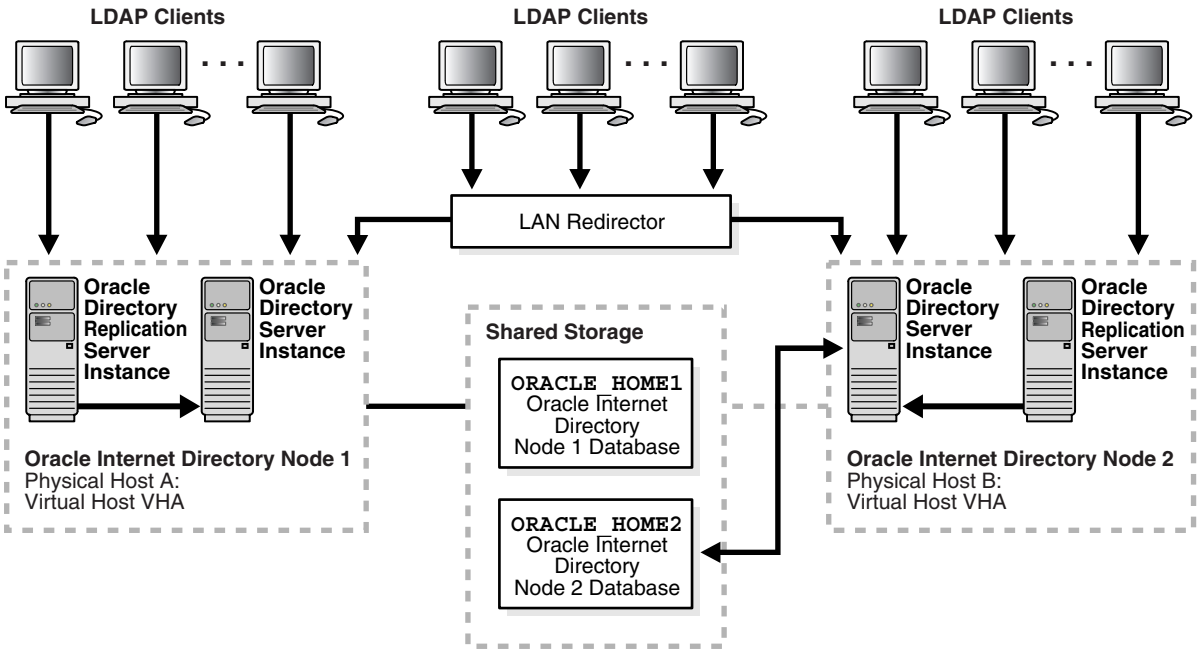
To automate the failover, you can write vendor-specific scripts to start the required processes. To effect transparent failover semantics, direct the cluster software to invoke the scripts.

After failover, LDAP clients continue to communicate with the same host as before, namely, the virtual host, VH. Consequently, the service disruption for these clients is minimal. The clients must reconnect when the failover is complete.

The Cold Failover Cluster Configuration in Conjunction with Oracle Internet Directory Replication

To provide additional availability and scalability, you can use the cold failover technique in conjunction with Oracle Internet Directory Replication. Figure 28–3 on page 28-6 illustrates this configuration.

Figure 28–3 Directory Replication in Conjunction with Cold Failover Configuration



As Figure 28–3 shows, on a two node cluster, Virtual Host VHA is hosted by Physical Host A and Virtual Host VHB is hosted by Physical Host B.

Oracle Internet Directory Node 1 is installed and configured on Virtual host VHA.

Oracle Internet Directory Node 2 is installed and configured on Virtual Host VHB.

Both Oracle Internet Directory nodes are configured for multimaster replication.

LDAP applications can do either of the following:

- Communicate directly with either Oracle Internet Directory node by using the respective virtual host names for the LDAP host
- Load-balance by means of a LAN re-director or another third-party solution that connects to the two hosts on which the Oracle Internet Directory nodes are configured

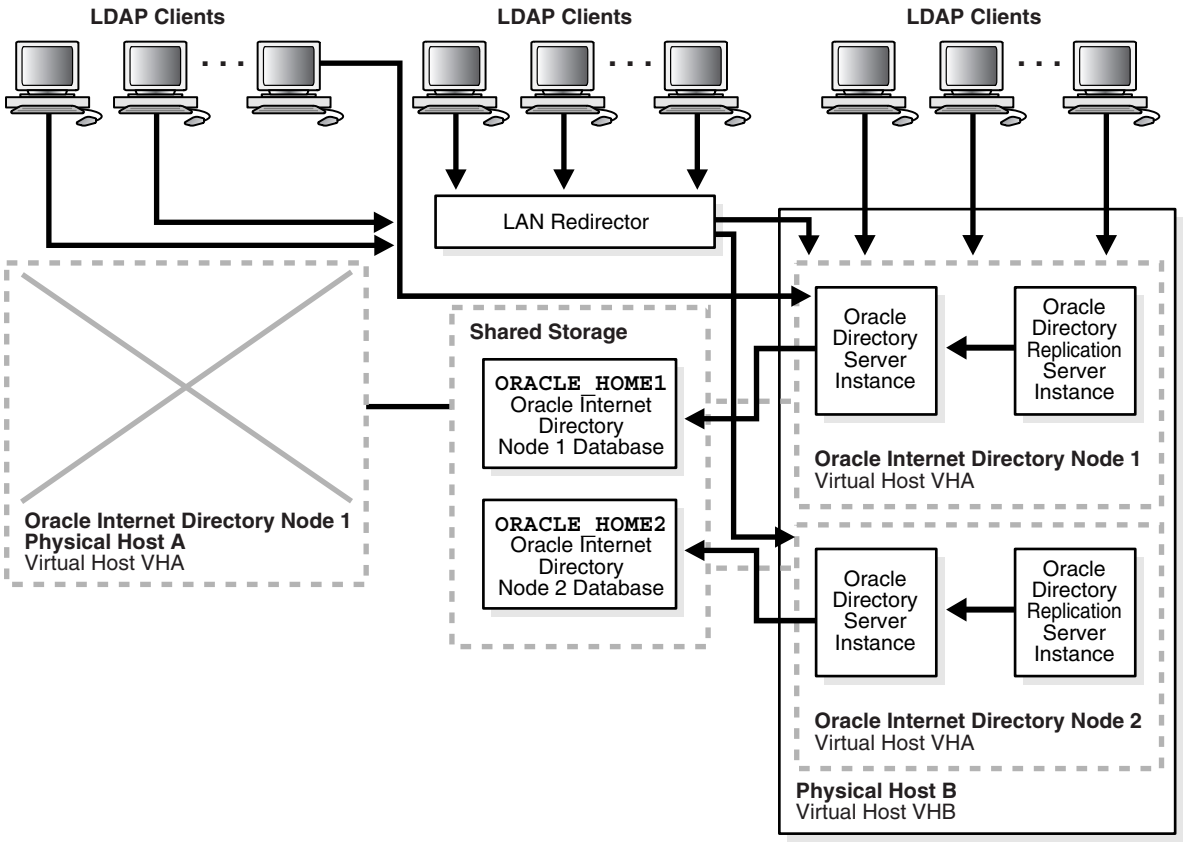
See Also: "An Oracle Internet Directory Node" on page 2-15

Using cold failover in this way represents an improvement over the simple cold failover configuration. There are two Oracle Internet Directory nodes and the two are in multimaster replication. Oracle Internet Directory is active on both cluster nodes and hence presents an active-active configuration. In contrast to the cold failover-only configuration, which is an active-passive configuration, the Oracle Internet Directory services are actively available on both cluster nodes at any given point in time.

The Cold Failover Process in Conjunction with Oracle Directory Replication

Figure 28–4 on page 28-8 shows the cold failover process in conjunction with Oracle directory replication.

Figure 28–4 The Cold Failover Process in Conjunction with Oracle Directory Replication.



As Figure 28–4 on page 28-8 shows, when Physical Host A fails or is unavailable because of maintenance downtime, the cluster software fails over virtual host VHA to Physical Host B. The Oracle Internet Directory processes that were previously running on Physical Host A are then restarted on Virtual Host VHA, and replication is resumed.

LDAP applications communicating directly with Oracle Internet Directory Node 1 by using host name VHA experience a momentary service outage. After the failover is complete, these applications must reconnect by using the same host name, namely, VHA. The momentary LDAP outage can be avoided completely if the two Oracle Internet Directory nodes are front-ended by a LAN redirector for load balancing.

29

The Directory 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 in which 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 a different node, that coordinate with each other 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

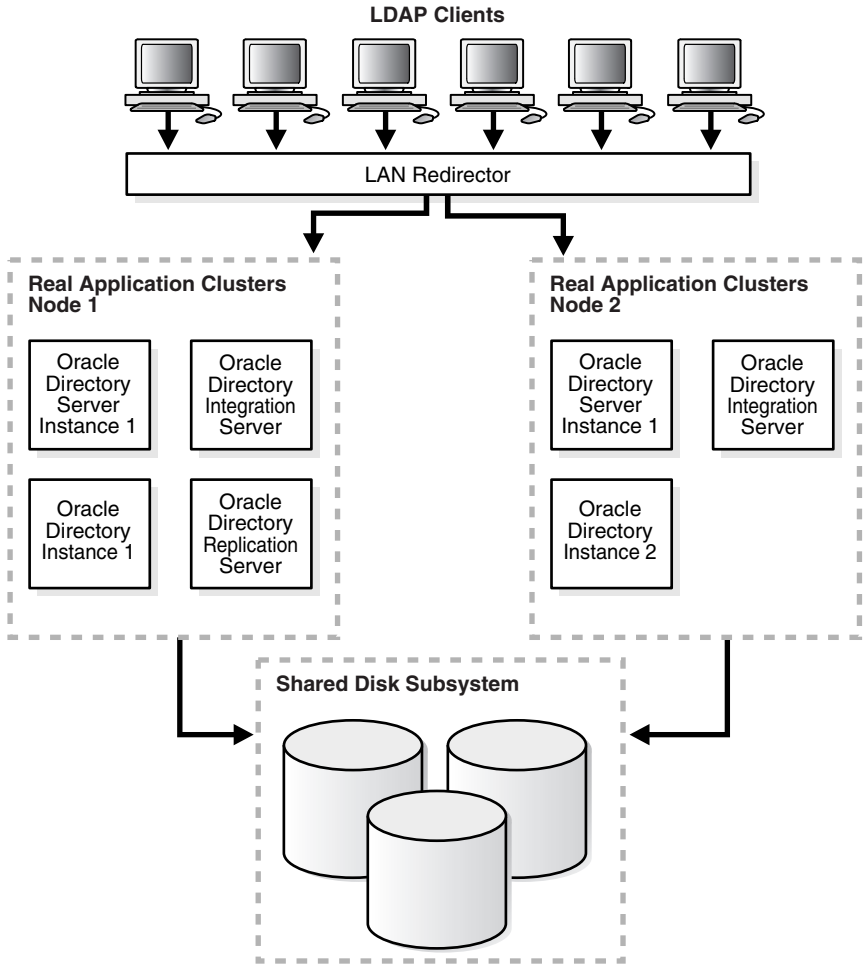
  Failover method in which a client connect request is forwarded to another listener if the first listener is not responding. It is enabled by service registration, because the listener knows whether an instance is running before attempting a connection.

The Oracle Directory Server in an Oracle9*i* Real Application Clusters Environment

To achieve a very comprehensive high availability configuration, you can configure Oracle Internet Directory to run in the Real Application Clusters active/active mode. This involves running Oracle Internet Directory processes and the Oracle Internet Directory-designated database on all the Real Application Clusters nodes.

Figure 29–1 shows a two-node cluster on which an Oracle9i Real Application Clusters database is configured.

Figure 29–1 Oracle Internet Directory with Basic High Availability Configuration



As Figure 29–1 shows:

- Oracle directory server instance 1 is active on Real Application Clusters Node 1 and Oracle directory server instance 2 is active on Real Application Clusters

Node 2. Note that multiple Oracle directory server instances can be started on each node.

- Oracle directory integration and provisioning server instances are active on both nodes.
- The Oracle directory replication server instance is active on one node only. If the node fails, then the OID Monitor on the surviving node pulls the Oracle directory replication server instance from the failed node and starts it on the surviving node.
- The LDAP client applications can be configured to communicate with Oracle Internet Directory on different Real Application Clusters nodes directly. Alternatively, the Oracle Internet Directory server instances can be front-ended by a LAN redirector to get a single system image of the Real Application Clusters nodes.
- When one Real Application Clusters node is unavailable because of failure or maintenance purposes, Oracle Internet Directory on the other Real Applications Clusters node is available. The LDAP clients connected to Oracle Internet Directory on the failed Real Applications Clusters node must reconnect.

Oracle Directory Server Connection Modes to Real Application Clusters Database Instances

This section discusses the various connection modes possible for Oracle directory server instances communicating with Oracle9*i* Real Application Clusters database instances. These connection modes are transparent to the Oracle Internet Directory clients, and do not affect the way in which Oracle Internet Directory communicates with its clients.

This section contains these topics:

Load_balance

If the `load_balance` parameter in the `tnsnames.ora` file is set to `on`, then Oracle Internet Directory connections to the Oracle9*i* Database Server is distributed to each

Oracle9*i* Database Server node. During failover of any node, only connections to the failed node are redirected to the available Oracle9*i* Database Server nodes.

If the `load_balance` parameter is set to `off`, then all the Oracle Internet Directory connections to the Oracle9*i* Database Server are to one Oracle9*i* Database Server node only.

During failover, all the connections are redirected to the available Oracle9*i* Database Server nodes.

Connect-Time Failover (CTF)

At the time of connection to the Oracle9*i* Database Server by the Oracle directory servers, if the primary Oracle9*i* Database Server node is not available, then Oracle Internet Directory servers connect to the backup—that is, secondary—database.

Transparent Application Failover (TAF)

To configure TAF, in the `tnsnames.ora` file, add `type=select` and `method=preconnect`.

During any LDAP search operation, if the primary Oracle9*i* Database Server node fails, then the Oracle directory server transparently connects to the backup—that is, the secondary—Oracle9*i* Database Servere node, and the current LDAP search operation continues.

Configuring the tnsnames.ora File for the Failover

This section shows configurations of the `tnsnames.ora` files on two nodes.

Node 1

```
db.us.acme.com=
 (description=
  (load_balance=off/on)  /* only connect time load balancing & connection load
balancing */
  (failover=on)          /* only connect time failover */
  (address=
       (protocol=tcp)
       (host=db1)
       (port=1521))
 (address=
       (protocol=tcp)
       (host=db2)
```

```
      (port=1521))
  (connect_data=
     (service_name=db.us.acme.com)
     (failover_mode=
       (backup=db2.acme.com)
       (type=select)
       (method=preconnect))))

db2.acme.com=
 (description=
  (address=
       (protocol=tcp)
       (host=db2)
       (port=1521))
  (connect_data=
     (service_name=db.us.acme.com)
     (instance_name=db2)
     (failover_mode=
     (backup=db2.acme.com)
     (type=select)
     (method=preconnect))
    ))
```

Node 2

```
db.us.acme.com=
 (description=
  (load_balance=off/on)  /* only connect time load balancing & connection load
balancing */
  (failover=on)          /* only connect time failover */
  (address=
       (protocol=tcp)
       (host=db2)
       (port=1521))
 (address=
       (protocol=tcp)
       (host=db1)
       (port=1521))
  (connect_data=
     (service_name=db.us.acme.com)
     (failover_mode=
       (backup=db1.acme.com)
       (type=select)
       (method=preconnect))))
```

```
db1.acme.com=
 (description=
  (address=
       (protocol=tcp)
       (host=db1)
       (port=1521))
  (connect_data=
     (service_name=db.us.acme.com)
     (instance_name=db2)
     (failover_mode=
     (backup=db2.acme.com)
     (type=select)
     (method=preconnect))))
```

Oracle Directory Replication Between Oracle Internet Directory Real Application Clusters Nodes

Directory replication can be configured between two or more Oracle Internet Directory Real Application Clusters nodes.

- Each node in the directory replication group (DRG) is an Oracle Internet Directory Real Application Clusters node
- Directory replication brings in geographic availability and the Oracle Internet Directory Real Application Clusters nodes in the DRG ensure local availability, manageability, and scalability

About Changing the ODS Password on a Real Application Clusters Node

If you change ODS password on one Real Application Clusters node by using the OID Database Password Utility, then you must update the wallet `$ORACLE_HOME/ldap/admin/oidpwdlldap1` on the other Real Application Clusters nodes. Do this either by copying the changed wallet to all the nodes, or by invoking the OID Database Password Utility on all other nodes to update the wallet file only. This applies to the replication password changes also. Here the Replication Environment Management Tool is used instead of the OID Database Password Utility.

Part VI

Delegation and Self-Service Administration in Oracle Internet Directory

This part contains these chapters:

30

Oracle Delegated Administration Services

This chapter describes Oracle Delegated Administration Services, a framework consisting of pre-defined, Web-based units for building administrative and self-service consoles. These consoles can be used by delegated administrators and users to perform specified directory operations.

It contains these topics:

About Oracle Delegated Administration Services

Oracle Delegated Administration Services is a set of pre-defined, Web-based units for performing directory operations on behalf of a user. It frees directory administrators from the more routine directory management tasks by enabling them to delegate specific functions to other administrators and to end users. It provides most of the functionality that directory-enabled applications require, such as creating a user entry, creating a group entry, searching for entries, and changing user passwords.

You can use Oracle Delegated Administration Services to develop your own tools for administering application data in the directory. Alternatively, you can use the Oracle Internet Directory Self-Service Console, a tool based on Delegated Administration Services. This tool comes ready to use with Oracle Internet Directory.

> **See Also:** Chapter 31, "Oracle Internet Directory Self-Service Console"

This section contains these topics:

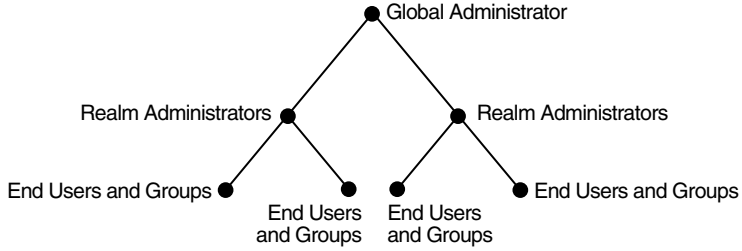
- How Oracle Delegated Administration Services Works
- Delegation of Directory Data Administration
- How Oracle Delegated Administration Services Provides Secure Access to the Directory

Delegation of Directory Data Administration

Applications built by using Oracle Delegated Administration Services enable you to grant a specific level of directory access to each type of user. For example, look at

Figure 30–1, which shows the various administrative levels in a hosted environment.

Figure 30–1 Administrative Levels in a Hosted Environment



The global administrator, with full privileges for the entire directory, can delegate to realm administrators the privileges to create and manage the realms for hosted companies. These administrators can, in turn, delegate to end users and groups the privileges to change their application passwords, personal data, and preferences. Each type of user can thus be given the appropriate level of privileges.

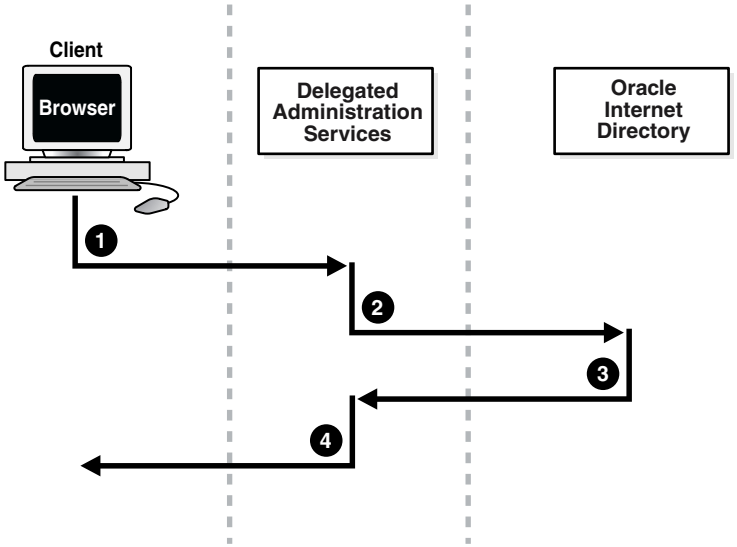
How Oracle Delegated Administration Services Works

Oracle Delegated Administration Services uses an Oracle Application Server Containers for J2EE (OC4J) that is enabled for small Java programs, called servlets. Together, the OC4J 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

Figure 30–2 shows the flow of information between components in a Oracle Delegated Administration Services environment.

Figure 30–2 Flow of Information Between Components in a Oracle Delegated Administration Services Environment



As Figure 30–2 shows:

1. The user, from a browser and using HTTP, sends to Oracle Delegated Administration Services a request containing a directory query.
2. Oracle Delegated Administration Services receives the request and launches the appropriate servlet. This servlet interprets the request, and sends it to Oracle Internet Directory by using LDAP.
3. Oracle Internet Directory sends the LDAP result to the Oracle Delegated Administration Services servlet.
4. The Oracle Delegated Administration Services servlet compiles the LDAP result into an HTML page, and sends it to the client Web browser.

How Oracle Delegated Administration Services Provides Secure Access to the Directory

When a user logs into an Oracle component, that component may need to obtain information from the directory on the end user's behalf—for example, the password verifier. To do this, the component typically logs into the directory as a **proxy user**, a feature that enables it to switch its identity to that of the end user.

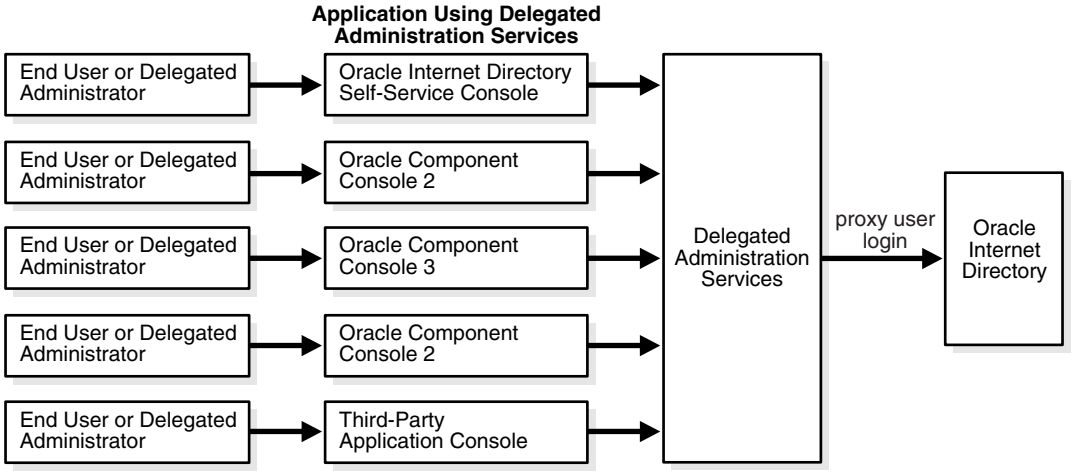
A problem, however, is that the greater the number of components logging into the directory as proxy users, the greater the risk of a malicious user accessing the directory as a proxy user. To prevent this security problem, the Oracle Delegated Administration Services centralizes proxy user access.

In a Oracle Delegated Administration Services environment, each component, instead of logging into the directory as a proxy user, logs into the central Oracle Delegated Administration Services. Oracle Delegated Administration Services then logs into the directory as a proxy user, switches its identity to that of the end user, and performs operations on that user's behalf. Centralizing proxy user directory access in this way replaces the less secure strategy of granting proxy user access to every component accessing the directory.

Figure 30–3 shows the proxy user feature in an Oracle Delegated Administration Services environment. End users or delegated administrators log in to a central Oracle Delegated Administration Services. They do this by using the Oracle Internet Directory Self-Service Console, the consoles of other Oracle components such as

OracleAS Portal, or those of third-party applications. The Oracle Delegated Administration Services then logs into Oracle Internet Directory as a proxy user.

Figure 30–3 Centralization of the Proxy User Feature in the Oracle Delegated Administration Services



Installing and Configuring Oracle Delegated Administration Services

This section tells you how to install and configure Oracle Delegated Administration Services. It contains these topics:

- Location of Log Files for Components in the Oracle Delegated Administration Services Environment
- Task 1: Install Oracle Delegated Administration Services
- Task 2: Verify that Oracle Delegated Administration Services Is Running
- Task 3: Configure the Default Identity Management Realm
- Task 4: Configure User Entries
- Task 5: Enable Debugging of Oracle Delegated Administration Services

Location of Log Files for Components in the Oracle Delegated Administration Services Environment

Table 30–1 tells you where to find the log files for components in the Oracle Delegated Administration Services environment.

Table 30–1 Log Files for Components In Oracle Delegated Administration Services Environment

Application	Log File Location
Oracle HTTP Server	*$ORACLE_HOME*/Apache/Apache/logs
Oracle Application Server Containers for J2EE (OC4J)	$*ORACLE_HOME*/j2ee/OC4J_SECURITY/log
Oracle Delegated Administration Services	$*ORACLE_HOME*/ldap/log/das.log
Oracle Process Manager (OPMN)	$*ORACLE_HOME*/opmn/logs

Task 1: Install Oracle Delegated Administration Services

Oracle Delegated Administration Services is installed as part of Oracle Internet Directory 10*g* (9.0.4).

Note: During installation,Oracle Delegated Administration Services is deployed in the OC4J_SECURITY instance. Because most of the Oracle Delegated Administration Services setup depends on this instance, its important that the name of this instance not be changed.

See Also: Oracle Application Server installation documentation for your operating system

Task 2: Verify that Oracle Delegated Administration Services Is Running

To verify that Oracle Delegated Administration Services 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 30–1 on page 30-7 to find log file locations for components in the Oracle Delegated Administration Services environment

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 30–1 on page 30-7 for the location of the log file

Step 3: Verify that the Oracle Application Server Single Sign-On Server Is Running

Using any browser, enter:

```
http://host_name:port_number/orasso/
```

where *host_name* is the name of the computer on which the Oracle HTTP Server is running, and *port_number* is the corresponding port number. The default port number of the Oracle HTTP Server is 7777. Try to log in by using the Oracle Application Server Single Sign-On login window.

Step 4: Verify that Oracle Delegated Administration Services 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, and *port_number* is the corresponding port number. The default port number of the Oracle HTTP Server is 7777. This displays the Oracle Delegated Administration Services home page.

Alternatively, you can verify that Oracle Delegated Administration Services is running by using the Enterprise Manager Web site. To do this:

1. On the Enterprise Manager Web site, navigate to the Instance Home Page, and scroll to the System Components section.
2. In the Name column, select OC4J_SECURITY. The home page for the component displays the status of Oracle Delegated Administration Services.

If Oracle Delegated Administration Services is not running, then start it by following the instructions in "Starting and Stopping Oracle Delegated Administration Services" on page 30-10.

Task 3: Configure the Default Identity Management Realm

To do this, follow the instructions in the section "Configuring an Identity Management Realm by Using the Oracle Internet Directory Self-Service Console" on page 31-11.

Task 4: Configure User Entries

To do this, follow the instructions in the section Configuring User Entries by Using the Oracle Internet Directory Self-Service Console on page 31-14.

Task 5: Enable Debugging of Oracle Delegated Administration Services

To enable or disable debugging of Oracle Delegated Administration Services, you modify the file `$ORACLE_HOME/ldap/das/das.properties`. Separate the values by using the vertical bar (|). After modifying this file, restart the Oracle Delegated Administration Services instance. Table 30–2 lists the default and possible values for the debug arguments.

Table 30–2 DAS.PROPERTIES File Debug Arguments

Flag	Default Value	Possible Values
DEBUG	FALSE	TRUE
		FALSE
DEBUG_LEVEL	none	ERROR
		SCHEMA
		TRACING
		SESSION

The DEBUG_LEVEL is interpreted only if the DEBUG flag is set to TRUE. TRACING is meant only for debugging purposes.

Starting and Stopping Oracle Delegated Administration Services

This section contains these topics:

- Starting and Stopping Oracle Delegated Administration Services by Using the Command Line
- Starting, Stopping, and Restarting Oracle Delegated Administration Services by Using Oracle Enterprise Manager

Starting and Stopping Oracle Delegated Administration Services by Using the Command Line

To start Oracle Delegated Administration Services by using the command line, enter:

```
ORACLE_HOME/dcm/bin/dcmctl start -co OC4J_SECURITY
```

To stop Oracle Delegated Administration Services by using the command line, enter:

```
ORACLE_HOME/dcm/bin/dcmctl stop -co OC4J_SECURITY
```

Starting, Stopping, and Restarting Oracle Delegated Administration Services by Using Oracle Enterprise Manager

To start, stop, or restart a component from the Enterprise Manager Web site:

1. On the Oracle Enterprise Manager Web site, navigate to the Instance Home Page, and scroll to the **System Components** section.
2. In the Name column, select **OC4J_SECURITY**. This opens the home page for the component.
3. In the **System Components** section, choose **Start**, **Stop**, or **Restart**.

See Also: "Task 2: Verify that Oracle Delegated Administration Services Is Running" on page 30-7

Creating Applications by Using Oracle Delegated Administration Services

You can embed Oracle Delegated Administration Services into both Oracle and third-party self-service applications that use Oracle Internet Directory. For example,

if you are building a Web portal, you can add Oracle Delegated Administration Services to enable end users to change application passwords stored in the directory.

Each unit has a corresponding URL stored in the directory. To invoke a Oracle Delegated Administration Services unit, an application queries the directory at runtime for the corresponding URL.

> **See Also:** The chapter on the Oracle Delegated Administration Services URL API in *Oracle Internet Directory Application Developer's Guide*

Oracle Delegated Administration Services for User Entries

Oracle Delegated Administration Services can perform these operations regarding user entries:

- Search for a user entry
- Create a user entry
- Self-edit a password
- Select a user entry and edit it
- Select a user entry and delete it
- Select a user entry and assign a privilege to that user
- View profile of the user who is logged in
- User list of values (LOV), a popup window that enables you to lookup and select a user
- Edit a user by passing the `orclguid` attribute to the URL. The entry is then displayed without the user needing to perform a search.
- Delete a user by passing the `orclguid` attribute to the URL. The entry is then displayed without the user needing to perform a search.
- Assign a privilege to a user by passing the `orclguid` attribute to the URL. The entry is then displayed without the user needing to perform a search.

Oracle Delegated Administration Services for Group Entries

Oracle Delegated Administration Services can perform these operations regarding group entries:

- Search for a group entry
- Create a group entry
- Select a group entry and edit it
- Select a group entry and delete it
- Select a group entry and assign a privilege to that group
- Group list of values (LOV), a popup window that enables you to lookup and select a group
- Edit a group by passing the `orclguid` attribute to the URL. The entry is then displayed without the user needing to perform a search.
- Delete a group by passing the `orclguid` attribute to the URL. The entry is then displayed without the user needing to perform a search.
- Assign a privilege to a group by passing the `orclguid` attribute to the URL. The entry is then displayed without the user needing to perform a search.

Configuring Oracle Delegated Administration Services by Using Oracle Enterprise Manager Application Server Control

You can use Oracle Enterprise Manager Application Server Control to configure Oracle Delegated Administration Services in the Oracle Identity Management infrastructure. When you do this, Enterprise Manager:

- Sets up the URL for Oracle Delegated Administration Services
- Configures the appropriate privileges
- Deploys Oracle Delegated Administration Services in an OC4J_SECURITY instance

To do this:

1. In Enterprise Manager, in the **Standalone Instances** section, choose the name of the Oracle Application Server instance. The corresponding screen for that instance appears.
2. Choose **Configure Components**. The Select Component screen appears.

3. Select **Oracle Delegated Administration Services**, then choose **Continue**. The Login screen appears.
4. Enter the user name and password of the directory super user. The default user name is `cn=orcladmin`.
5. Choose **Finish** to complete the configuration.
6. Start the OC4J_SECURITY instance where Oracle Delegated Administration Services is deployed. To do this:
   a. In the **System Components** section, select **OC4J_SECURITY**.
   b. Choose **Start**.

Manually Deploying Oracle Delegated Administration Services

Oracle Delegated Administration Services are deployed automatically in the infrastructure installation of Oracle Application Server. In some situations, you may need to deploy it on a computer other than that on which the infrastructure is deployed. To do this, follow these steps:

1. Verify that the computer has at least the core installation installed and the installation is pointing to an existing Oracle Internet Directory/Oracle Application Server Single Sign-On.
2. Navigate to the *ORACLE_HOME*`/dcm/bin` directory.
3. Create a new component using the following command:

   ```
   dcmctl createcomponent -verbose -debug -ct oc4j -co OC4J_SECURITY
   ```

4. Start the component by using the following command:

   ```
   dcmctl start -verbose -debug -co OC4J_SECURITY
   ```

5. Deploy the `oiddas.ear` file using the following command:

   ```
   dcmctl deployApplication -debug -verbose -a oiddas -f
   ORACLE_HOME/ldap/das/oiddas.ear -co OC4J_SECURITY
   ```

6. Perform the following steps to add the `LD_LIBRARY_PATH` and `DISPLAY` environment variables to the opmn.xml file:
   a. Navigate to the *ORACLE_HOME*`/opmn/conf` directory and open `opmn.xml` in a text editor.
   b. Add the following lines in the OC4J_SECURITY section of opmn.xml:

For a UNIX environment:

```
<environment>
<prop name="DISPLAY" value="%hostname%:0.0"/>
<prop name="LD_LIBRARY_PATH" value="%ORACLE_HOME%/lib"/>
</environment>
```

For a Windows environment:

```
<environment>
<prop name="PATH" value="%ORACLE_HOME%/bin"/>
</environment>
```

Replace hostname and ORACLE_HOME with the appropriate values. Hostname should point to a computer where X server is running.

Note the placement of the section `<environment>` in the following example.

```
<oc4j maxRetry="3" instanceName="OC4J_DAS" gid="OC4J_SECURITY"
numProcs="1">
<config-file path="/home/ias902/j2ee/OC4J_
DAS/config/server.xml"/>
<oc4j-option value="-properties"/>
<port ajp="3001-3100" jms="3201-3300"
rmi="3101-3200"/>
<environment>
<prop name="DISPLAY" value="sandal:0.0"/>
<prop name="LD_LIBRARY_PATH" value="/home/ias902/lib"/>
</environment>
</oc4j>
```

c. Navigate to the *ORACLE_HOME*`/dcm/bin` directory.

d. Save the changes to the repository by using the following command:

```
dcmctl updateconfig -verbose -debug -ct opmn
```

e. Restart OPMN by using the following command:

```
dcmctl restart -verbose -ct opmn
```

f. Stop and start the OC4J_SECURITY instance by using the following commands:

```
dcmctl stop -verbose -debug -ct oc4j -co OC4J_SECURITY
dcmctl start -verbose -debug -ct oc4j -co OC4J_SECURITY
```

g. Set the necessary permissions for Oracle Delegated Administration Services. Modify the group by using either Oracle Directory Manager or the command-line tool. Add the DN of the new Oracle Application Server instance where Oracle Delegated Administration Services is currently being deployed as the `uniquemember`.

```
DN of the group to be modified:
cn=Associated
Mid-tiers,orclApplicationCommonName=DASApp,cn=DAS,cn=Products,cn=OracleC
ontext
```

The DN on the Oracle Application Server instance is:

```
orclApplicationCommonName=name of Oracle Application Server
instance,cn=IAS Instances, cn=IAS, cn=Products,
cn=OracleContext
```

where *`name of Oracle Application Server instance`* is obtained from `$ORACLE_HOME/config/ias.properties`.

31

Oracle Internet Directory Self-Service Console

This chapter describes the Oracle Internet Directory Self-Service Console, a ready-to-use application created by using Oracle Delegated Administration Services.

It contains these topics:

Delegated Administration Through the Oracle Internet Directory Self-Service Console

This section contains these topics:

About Delegated Administration

Delegated administration, provided through the Oracle Identity Management infrastructure, enables you to store all data for users, groups, and services in a central directory, while distributing the administration of that data to various administrators and end users. It does this in a way that respects the various security requirements in your environment.

For example, your enterprise may require one administrator for user data, and another for the e-mail service. Or it may require the administrator of a component like Oracle Financials to have full control of user privileges, and that of another component like OracleAS Portal to have full control of the Web pages for a specific user or group. Delegated administration, as provided by the Oracle Identity Management infrastructure, enables all of these administrators with their diverse security requirements to administer the centralized data in a way that is both secure and scalable. Within this environment, the Oracle Internet Directory Self-Service Console is a convenient tool for delegating administration to various administrators or to end users.

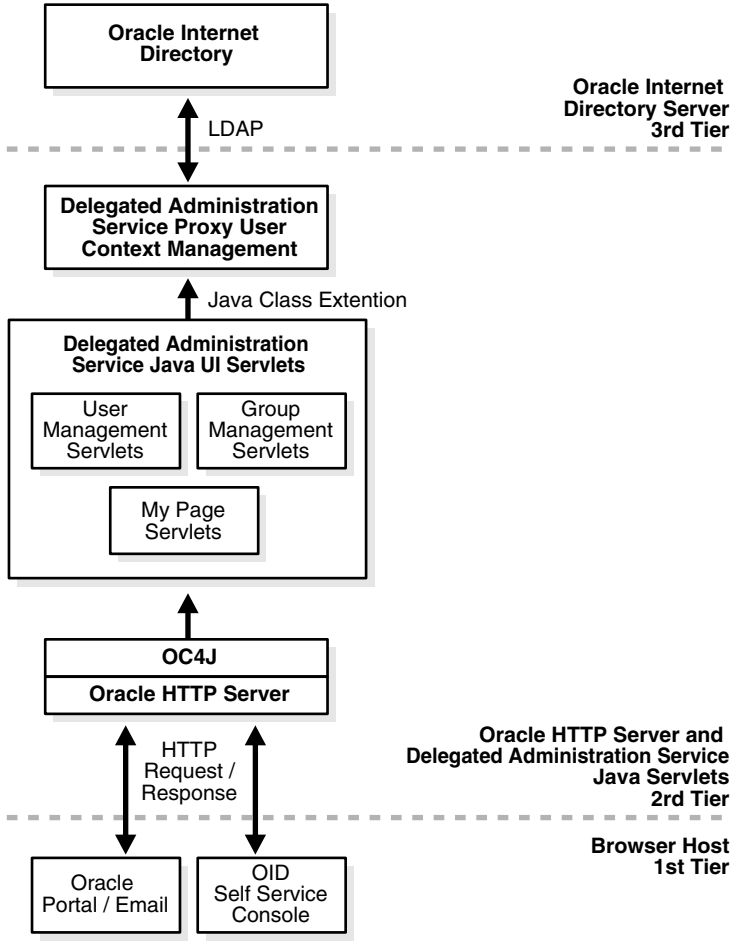
> **See Also:** Chapter 17, "Delegation of Privileges for an Oracle Technology Deployment" for more information about delegated administration

About the Oracle Internet Directory Self-Service Console

The Oracle Internet Directory Self-Service Console enables you to delegate administrative privileges to various administrators and to end users. It is a ready-to-use application created by using Oracle Delegated Administration Services. It provides a single graphical interface for delegated administrators and end users to manage data in the directory.

Figure 31–1 shows how the Oracle Internet Directory Self-Service Console interacts with Oracle Delegated Administration Services.

Figure 31–1 Interactions of Oracle Delegated Administration Services Components



Using the Oracle Internet Directory Self-Service Console

The Oracle Internet Directory Self-Service Console enables both administrators and users, depending on their privileges, to perform various directory operations. This section contains these topics:

- Getting Started with the Oracle Internet Directory Self-Service Console
- Searching for Entries by Using Oracle Internet Directory Self-Service Console
- Performing the Tasks of an End User
- Performing the Tasks of an Administrator

Getting Started with the Oracle Internet Directory Self-Service Console

This section explains how to start, log in to, and stop the Oracle Internet Directory Self-Service Console.

Starting and Stopping the Oracle Internet Directory Self-Service Console

To use the Oracle Internet Directory Self-Service Console, you need to start the Oracle Delegated Administration Services if it is not already running.

> **See Also:** "Starting and Stopping Oracle Delegated Administration Services" on page 30-10 for instructions on starting Oracle Delegated Administration Services

Logging into the Oracle Internet Directory Self-Service Console

To log in to the Oracle Internet Directory Self-Service Console:

1. Visit the URL of the Oracle Internet Directory Self-Service Console.
2. In the upper right corner, select **Login**. This takes you to the Oracle Application Server Single Sign-On window.
3. In the Single Sign-On window, in the **User Name** field, enter your Self-Service Console user name--for example, `jdoe`.
4. In the **Password** field, enter your Self-Service Console password.
5. If you are in a hosted environment in which there are multiple hosted companies, then the **Company** field appears. Otherwise, it does not appear. If the **Company** field appears, then enter the name of your company.
6. Choose **Login**.

Searching for Entries by Using Oracle Internet Directory Self-Service Console

The Oracle Internet Directory Self-Service Console enables you to search for both user and group entries.

Searching for User Entries by Using the Oracle Internet Directory Self-Service Console

To search for user entries:

1. In the Oracle Internet Directory Self-Service Console, select the **Directory** tab, then select **Users**.
2. In the **Search for User** field, enter the first few characters of one of the following:
   - First name
   - Last name
   - Login name
   - The e-mail identifier
   - The `cn` attribute of the user

   For example, if you are searching for Anne Smith, you could enter `Ann` or `Smi`.

   To generate a list of all users in the directory, leave this field blank.
3. Choose **Go** to display the search results.

Searching for Group Entries by Using the Oracle Internet Directory Self-Service Console

To search for a group entry:

1. Select the **Directory** tab, then select **Groups**.
2. In the **Search Group Name** text box, enter the first few characters of the name of the group for which you are searching.

   To generate a list of all groups in the directory, leave this field blank.
3. Choose **Go** to display the entries that match the criteria you entered.

Performing the Tasks of an End User

This section tells you, as an end user, how to configure and modify elements in your personal profile, including password, photo, time zone, and resource access information. Table 31–1 lists the administrative tasks, and points you to the corresponding information.

Table 31–1 Tasks of an End User

Task	Where to Find Instructions
Editing your profile	"Editing Your Profile" on page 31-6
Changing your own password	"Changing Your Own Password and Password Hint" on page 31-6
Resetting your password	"Resetting Your Password If You Forget It" on page 31-8
Viewing your organization chart	"Viewing Your Organizational Chart" on page 31-8
Changing time zone settings	"Changing Your Time Zone Setting" on page 31-8
Configuring resource access information	"Managing Resource Access Information" on page 31-9

Editing Your Profile

To edit your profile:

1. Select the **My Profile** tab page, then choose **Edit My Profile**. The Edit My Profile window appears.
2. Make your changes.
3. Choose **OK**.

> **Note:** To refresh the My Profile tab page with the latest information in the server, choose Refresh My Profile. Do not use the refresh or reload button on your browser, which simply refreshes with information from the mid-tier cache and not from the server.

Changing Your Own Password and Password Hint

You can use the Self-Service Console to change your own password to Oracle Application Server Single Sign-On and other Oracle components. Changing your password for Oracle Application Server Single Sign-On also changes your password

for any applications that use Oracle Application Server Single Sign-On for authentication.

To change your password, select the **My Profile** tab, then select **Change My Password**. This displays the Change My Password window. You can use this window to change your password to either Oracle Application Server Single Sign-On or to another Oracle component.

To change your password to Oracle Application Server Single Sign-On:

1. In the **Single Sign-On** section, in the **Old Password** field, enter your current password.
2. In the **New Password** field, enter your new password, then confirm it by entering it again in the **Confirm New Password** field.
3. In the **Password Hint** field, enter a question—for example, your mother's maiden name. If you later forget your password, then you will be asked this question. If your answer is correct, then your password will be retrieved for you.
4. In the **Answer to Password Hint** field, enter the answer to the hint you just entered in the previous field.
5. Choose **Submit**.

> **Note:** When you enter an answer to your password hint in the Answer to Password Hint field, be sure to remember the answer exactly as you entered it. Any deviation—for example, extra spaces, additional hyphens, or capitalizations—cause the hint answer not to match the stored version.

To change your password to another Oracle component that is not enabled for Oracle Application Server Single Sign-On:

1. In the **Application Passwords** section, select the Oracle component for which you want to specify a new password.
2. Choose **Update Password**. This displays the **Change Application Password** window.
3. In the **New Password** field, enter your new password, then confirm it in the **Confirm New Password** field.
4. Choose **Submit**.

Resetting Your Password If You Forget It

If you forget your password, you can reset it. For security reasons, this requires you to answer the question you specified when you first established your password.

1. In the Oracle Internet Directory Self-Service Console home page, choose **Forgot My Password**. The Reset My Single Sign-On page appears.
2. In the **Confirm Identity** section, enter values for the fields. These fields are specific to your environment and are configured by the administrator. You must also enter the name of your company.
3. Choose **Next**. The Confirm Additional Personal Information window appears.
4. If, in "Changing Your Own Password and Password Hint" on page 31-6, you set your password hint, then the Confirm Additional Personal Information window asks you a question based on that hint. Enter the answer to the password hint you specified in that step.

   If you did not previously set a password hint, then the Confirm Additional Personal Information window prompts you for other personal data as configured by your administrator. This data is then used to validate your identity.
5. Choose **Next**. The Reset SSO Password window appears.
6. In the **New Password** field, enter your new password, then confirm it by entering it again in the **Confirm New Password** field.
7. Choose **Submit**.

Viewing Your Organizational Chart

To locate yourself within the hierarchy of your organization, you can view your organization chart.

To view your organization chart, select the **My Profile** tab, then choose **View My Org Chart**.

Changing Your Time Zone Setting

To change your time zone setting this:

1. Select the **My Profile** tab, then select **Change My Time Zone**. This takes you to the Time Zone Settings window.
2. In the Time Zones Settings window, select your new time zone, then choose **Submit**.

Managing Resource Access Information

You can use the Oracle Internet Directory Self-Service Console to create, modify, and delete resource access information.

> **See Also:** "Resource Information" on page 2-36 for a discussion of resource access information

---

Note: The **Preferences** link mentioned in the following procedures appears only if the administrator has created resource access information for the user.

Creating Resource Access Information To specify resource access information:

1. Select the **My Profile** tab, then choose **Preferences**.
2. Choose **Create**. The Create Resource window appears.
3. In the **Resource Name** field, specify the name of the resource or service to be accessed by the component on your behalf.
4. From the **Resource Type** list, select the type of resource to be accessed. Default options are:
   - **OracleDB**: an Oracle9*i* Database Server
   - **ExpressPDS**: Oracle Express Pluggable Data Source
   - **JDBCPDS**: Java Database Connectivity Pluggable Data Source

   Other resource types may appear in this list as specified by the administrator.
5. Choose **Next**. The Resource Access Information window appears.
6. In the Resource Access Information window, enter the appropriate information.
7. Choose **Submit**.

Modifying Resource Access Information To modify resource access information:

1. Select the **My Profile** tab, then choose **Preferences**.
2. Select the resource whose information you want to modify, then choose **Edit**. The Edit Resource window appears.
3. In the Edit Resource window, enter the appropriate information.
4. Choose **Submit**.

Deleting Resource Access Information To delete resource access information:

1. Select the **My Profile** tab, then choose **Preferences**.
2. Select the resource whose information you want to delete.
3. Choose **Delete**.

> **See Also:** "Resource Information" on page 2-36 for a brief description of resource access information

Performing the Tasks of an Administrator

As an administrator, you can perform all of the tasks of an end user, as well as those for which you have the necessary administrative privileges. Table 31–2 lists the administrative tasks, and points you to the corresponding information.

Table 31–2 Tasks of an Administrator

Task	Where to Find Instructions
Managing identity management realms	"Configuring an Identity Management Realm by Using the Oracle Internet Directory Self-Service Console" on page 31-11
	"Creating an Additional Identity Management Realm by Using the Oracle Internet Directory Self-Service Console" on page 31-14
Managing user entries	"Configuring User Entries by Using the Oracle Internet Directory Self-Service Console" on page 31-14
	"Creating User Entries by Using the Oracle Internet Directory Self-Service Console" on page 31-17
	"Modifying User Entries by Using the Oracle Internet Directory Self-Service Console" on page 31-18
	"Deleting User Entries by Using the Oracle Internet Directory Self-Service Console" on page 31-19
	"Assigning Privileges to Users by Using the Oracle Internet Directory Self-Service Console" on page 31-19
	"Changing the Password of a User by Using the Oracle Internet Directory Self-Service Console" on page 31-20

Table 31–2 (Cont.) Tasks of an Administrator

Task	Where to Find Instructions
Managing group entries	"Creating Group Entries by Using the Oracle Internet Directory Self-Service Console" on page 31-20
	"Modifying Group Entries by Using the Oracle Internet Directory Self-Service Console" on page 31-22
	"Deleting Group Entries by Using the Oracle Internet Directory Self-Service Console" on page 31-22
	"Assigning Privileges to Groups by Using the Oracle Internet Directory Self-Service Console" on page 31-22
Managing services	"Modifying Service Properties" on page 31-23
	"Modifying Subscription Information for a Service Recipient" on page 31-23
Managing accounts	"Managing Accounts" on page 31-24
Managing resource access information	"Configuring Resource Type Information" on page 31-25
	"Creating User Entries by Using the Oracle Internet Directory Self-Service Console" on page 31-17
	"Configuring Default Resource Access Information" on page 31-26

Configuring an Identity Management Realm by Using the Oracle Internet Directory Self-Service Console

If you have the administrative privileges, then you can specify the following for an identity management realm:

- The attribute by which you want users to identify themselves when they log in
- The root entries of the user search base and of the group search base—that is, the locations in the directory information tree containing entries for users and groups
- The root entries for the user creation base and the group creation base—that is, the location in the DIT where users and groups are created. This can be the same as the user search base, or it can be a location under the user search base.
- The display of realm and product logos

How you modify an identity management realm entry depends on whether you are the administrator for a service provider or the administrator for a hosted company.

If You Are the Administrator for a Hosted Company To configure the identity management realm of the hosted company for which you are the administrator:

1. Select the **Configuration** tab.
2. In the Identity Management Realm window, enter values for the various fields. These fields are described in Table C–48 on page C-46.
3. Choose **Submit** to save your changes.

Note:

- The value you enter in the Naming Attribute field should not be the same as the value you enter in the Attribute for Login Name field.
- Although you can enter more than one value in the User Search Base field, doing so can degrade performance.

If You Are the Global Administrator or the Administrator for a Service Provider The Oracle Internet Directory Self-Service Console enables you to configure all of the information the realm administrator for a hosted company can configure. You can also create entries for hosted company realms and provision applications and services in the application services provider environment.

To configure an identity management realm:

1. At the top right of the Oracle Internet Directory Self Service Console, choose the **Realm Management** icon. This displays the Identity Management Realm window.
2. In the Identity Management Realm window, in the **Search Identity Management Realm** field, enter all or part of the name of the realm whose entry you want to modify, then choose **Go**. This displays a list of realms that match your search criteria.
3. From the search results list, select the realm you want to modify, then choose **Proceed**. This takes you to the identity management realm you want to modify.
4. Select the **Configuration** tab. In the Identity Management Realm window, enter values in the appropriate fields. These fields are described in Table C–48 on page C-46.
5. Choose **Submit**.

Configuring the Parent DN for Entries You can specify one or more parent DNs for entries. If you specify more than one, then a delegated administrator can choose the one under which to place a new user entry.

There are two ways to specify parent DNs. The first is by specifying values for the User Creation Base, and the second is by specifying values for the organizational units (`ou`) attribute. If you specify a different value for each, then those for the `ou` attribute prevail.

To specify parent DNs by providing values for the User Creation Base:

1. Select the **Configuration** tab, then select **Identity Management Realm**.
2. In the **User Creation Base** field, enter one or more DNs, one line for each DN.
3. Choose **Submit**.

Alternatively, you can specify parent DNs by setting the value for the organizational unit (`ou`) attribute. If you do this, then a delegated administrator can choose the organization unit under which to place user entries. To specify a parent DN by using this method:

1. Select the **Configuration** tab, then select **User Entry**.
2. Choose **Next**. The Configure User Attributes window appears.
3. Choose Add New Attribute. The Add New Attribute window appears.
4. From the **Directory Attribute Name** list, select the `ou` attribute.
5. From the **UI Type** list, select **Predefined List**.
6. In the **LOV Values** field, enter the display name of the parent DN, followed by three semicolons (;), followed by the DN itself. You can add more parents DNs, one line for each.

   For example:

   ```
   Sales;;;cn=users,dc=us,dc=my_company,dc=com
   HR;;;cn=groups,dc=us,dc=my_company,dc=com
   ```

Following this example, when a delegated administrator chooses the organizational unit under which to place a user entry, she selects from a list displaying `Sales` and `HR`.

Creating an Additional Identity Management Realm by Using the Oracle Internet Directory Self-Service Console

If you have the administrative privileges, then you create an entry for an identity management realm as follows:

1. At the top right of the Oracle Internet Directory Self Service Console, choose the **Realm Management** icon. This displays the Identity Management Realms window.
2. In the Identity Management Realms window, choose **Create**. The Create Identity Management Realm window appears.
3. In the Create Identity Management Realm window, enter the appropriate values in the fields. These fields are described in Table C–47 on page C-45.
4. Choose **Submit**.

Configuring User Entries by Using the Oracle Internet Directory Self-Service Console

When a user creates or edits a user entry, the Oracle Internet Directory Self-Service Console displays various categories—including, for example, basic information, password, and photo—each with its own set of attributes. You can specify which of these categories the console displays, and how it displays them and their corresponding attributes.

Specifically, the Oracle Internet Directory Self-Service Console enables you to:

- Associate object classes with user entries, and add and modify these object classes
- Specify the categories of attributes you want to enable users to add or modify
- Customize the way the Oracle Internet Directory Self-Service Console displays those categories and attributes

To configure user entries:

1. Select the **Configuration** tab, then select **User Entry**. This displays the Configure User Object Classes window listing the existing object classes associated with user entries.

2. To add an object class for user entries:

   a. In the Configure User Object Classes window, 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 Configure Object Class window. The object class you just chose is now listed as an object class for user entries.

   c. To add more object classes, repeat these steps, or, to move to the next step, choose **Next** to display the Configure User Attributes window.

3. The Configure User Attributes window lists some—but not all—of the attributes of the object classes you specified in Step 2 on page 31-15. There may be other attributes belonging to those object classes as well. You can add as many of those other attributes as you wish by following the instructions in this step. You can modify how the attributes are displayed or delete attributes.

   To add attributes to user entries:

   a. In the Configure User Attributes window, choose **Add New Attribute**. This displays the Add New Attribute window.

   b. In the Add New Attribute window, enter values for the fields. These are described in Table C–44 on page C-42.

   c. Choose **Done**. This returns you to the Configure User Attributes window. The attribute you just chose is now listed in the Attribute Configuration list.

   d. To add more attributes, repeat these steps.

   To modify the display of attributes:

   a. In the Configure User Attributes window, in the **Directory Attribute Name** column, select the attribute you want to modify, then choose **Edit**. This displays the Editing Attribute window.

   b. In the Editing Attribute window, enter values for the fields. These are described Table C–45 on page C-43.

   c. Choose **Done**. This returns you to the Configure Attributes window. The attribute configurations you just made are now reflected in the Directory Attribute Name list.

   d. To configure or modify more attributes, repeat these steps.

   To delete attributes of user entries, in the Configure User Attributes window, in the **Directory Attribute Name** list, select the attribute you want to configure, then choose **Delete**.

4. To customize the display of categories, in the Configure User Attributes window choose **Next** to display the Configure Attribute Categories window. This window contains a table listing the existing categories, the name displayed to the user, and the display order of each category.

   a. To add a new category, choose **Create**. This displays the Create window. In the **UI Label** field, enter the name of the category as you would like it displayed in the interface.

   b. To modify the display name of a category, in the **UI Label** column, edit the field for each attribute you want to modify.

   c. To set the display order of categories, choose **Order**. The Order window displays the various categories you just specified. Use the up and down arrows to move the categories into the desired order.

   d. To set the display order of attributes for each category, select the category, then choose **Edit**. In the Order window, use the arrow buttons to set the display order of the attributes, or to remove an attribute from being displayed.

   e. To delete a category, select the category, then choose **Delete**.

   When you have finished configuring attribute categories, choose **Next** to display the Configure Search Table Columns window.

5. When a user performs a search, the results are displayed in a table. You can specify the number of columns in that table and their headings. To configure search table columns:

   a. In the **All Attributes** box, select one or more attributes that you want to be represented in the search results. These will serve as column headings in the search results table.

   b. Use the left-right arrows to move the attributes to the **Selected Attributes** box.

   c. In the **Selected Attributes** box, order the attributes by using the up-down arrows to the right of the box. The first attribute in the list represents the column farthest to the left in the search results table.

   When you have finished configuring the search results table, choose **Next** to display the Configure Roles window.

6. To enable users to assign roles to users, in the **Enable Roles** category, select "Enable Role assignment in the user management interface".

   You can specify the roles that users can assign other users.

To add a role that users can assign other users:

a. Choose **Add Role** to display the Search and Select: Roles window.

b. In the **Group Name Begins With** field, enter the first few letters of the name of the administrative group you want to add.

c. From the search results, select the name of the administrative group you want to add, then choose **Select**. This returns you to the Configure Roles window. The administrative group you just selected appears in the Roles list.

To delete a role, select it from the table and choose **Delete**.

7. When you have finished configuring user entries, choose **Finish**.

Creating User Entries by Using the Oracle Internet Directory Self-Service Console

To create a user entry:

1. Select the **Directory** tab, then select **User Entry**.
2. Choose **Create** to display the Create User window.
3. In the Create User window, some of the sections are unique to your environment, others are integral to the Oracle Internet Directory Self-Service Console. The latter are:

   - **Roles Assignment**, which enables you to assign one or more roles to this user
   - **Resource Access Information**, which enables you to grant this user access to resources specific to Oracle Forms and Oracle Reports.

   Enter values in the fields unique to your environment.

   To enter values for fields that are integral to Oracle Internet Directory Self-Service Console:

   In the **Roles Assignment** section, in the **Select** column, select the role that you want to assign to this user.

   In the **Resource Access Information** section, in the **Select** column, select the resource to which you want this user to have access. If no resource access information has been specified, then you can create it. To do this:

   a. In the **Resource Access Information** section, choose **Create**. The Create Resource window appears.

b. In the **Resource Name** field, specify the name of the resource or service to be accessed by the component on your behalf.

c. From the **Resource Type** list, select the type of resource to be accessed. Default options are:

   * **OracleDB**: an Oracle9*i* Database Server
   * **ExpressPDS**: Oracle Express Pluggable Data Source
   * **JDBCPDS**: Java Database Connectivity Pluggable Data Source

   Other resource types may appear in this list as specified by the administrator.

d. Choose **Next**. The Resource Access Information window appears.

e. In the Resource Access Information window, enter the appropriate information.

f. Verify that you have entered all information correctly, then choose **Submit**.

Modifying User Entries by Using the Oracle Internet Directory Self-Service Console

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. In the Edit User window, some of the sections are integral to the Oracle Internet Directory Self-Service Console, while others are unique to your environment. The sections integral to the Oracle Internet Directory Self-Service Console are:

   - **Roles Assignment**, which enables you to assign one or more roles to this user
   - **Resource Access Information**, which enables you to create, modify, or delete resource access information
   - **Existing Group Memberships**, which displays the groups of which this user is already a member
   - **Edit History**, which tells you who created or modified this user entry, and when the entry was created or modified

To enter values for fields that are integral to Oracle Internet Directory Self-Service Console:

a. In the **Role Assignment** section, in the **Select** column, select the role that you want to assign to this user.

b. In the **Resource Access Information** section, in the **Select** column, select the resource to which you want this user to have access.

After you have entered information in the fields that are integral to Oracle Internet Directory Self-Service Console, do the same for the fields unique to your environment.

4. Choose **Submit**.

Deleting User Entries by Using the Oracle Internet Directory Self-Service Console

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 Oracle Internet Directory Self-Service Console

You can privilege a user to:

- Create, edit, and delete users and groups
- Assign privileges to other users and 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 entry of the user to whom you want to assign privileges.
2. From the search results list, select the user to whom you want to assign privileges, then choose **Assign Privilege**. The Assign Privileges to User window displays a list of privileges.
3. Select the privileges you want to assign to this user. These are described in Table C–46 on page C-44.

4. Choose **Submit**, or, to assign privileges to another user, choose **Specify Other User** and repeat the process.

Changing the Password of a User by Using the Oracle Internet Directory Self-Service Console

If you have the necessary access rights, you can change the password of a user other than yourself. To change another user's password:

1. Select the **Directory** tab, then select **Users**.
2. Perform a search for the entry of the user whose password you want to change.
3. From the results of your search, select the user entry, then choose **Edit** to display the Edit User window.
4. In the **Basic Information** section, enter and confirm the password you want to assign to the user.
5. Choose **Submit**.

Note: If you do not have the privileges to edit a user entry, then the Edit button does not appear, and you cannot perform this operation.

Creating Group Entries by Using the Oracle Internet Directory Self-Service Console

To create a group entry:

1. Select the **Directory** tab, then select **Group**.
2. Choose **Create**. This displays the Create Group window.
3. In the Create Group window, in the **Basic Information** section, in the **Name** field, enter the name for this group.
4. In the **Display Name** field, enter the friendly name for this group. For example, if the **RDN** is `OracleDBCreators`, then you could enter the display name as `Oracle Database Creators`.
5. Optionally, in the **Description** field, enter a brief description of this group.
6. To hide this group entry from all but its owners, in the **Group Visibility** field, select **Private**. Otherwise, accept the default, namely, **Public**.

7. Configure owners of this group. Note that the creator of the group is automatically a group owner.

   To add a user as an owner of this group:

   a. In the **Owners** section, choose **Add User**. This displays the Search and Select: User window.

   b. Search for the entry of the user you want to add as an owner of the group.

   c. Choose **Select**. This returns you to the Create Group window. The user you specified is listed in the **Owners** section.

   To add a group as an owner of this group:

   a. In the **Owners** section, choose **Add Group**. This displays the Search and Select: Group window.

   b. Search for the entry of the group you want to add as an owner of the group.

   c. Choose **Select**. This returns you to the Create Group window. The group you specified is listed in the **Owners** section.

   To remove a user or group as an owner of this group, select the user or group, then choose **Remove**.

8. Configure members of this group.

   To add a user as a member of this group:

   a. In the **Members** section, choose **Add User**. This displays the Search and Select window.

   b. Search for the entry of the user you want to specify as a member of this group.

   c. Choose **Select**. This returns you to the Create Group window. The user you specified is listed in the **Members** section.

   To remove a user from this group, in the **Members** section, select the user's name and choose **Remove**.

   To add a group as a member of this group:

   a. In the **Members** section, choose **Add Group**. This displays 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.

9. You can assign roles to this group.

   To specify the roles that you want to assign to this group, in the **Roles Assignment** section, in the **Select** column, select the role that you want to assign to this group.

   To remove the role from the group, in the **Roles Assignment** section, in the **Select** column, deselect the role that you want to remove from this group.

Modifying Group Entries by Using the Oracle Internet Directory Self-Service Console

To modify a group entry:

1. Select the **Directory** tab and perform a search for the group entry you want to modify.
2. From the search results, select the group entry you want to modify.
3. Choose **Edit**. This displays the Edit Group window.
4. Modify the fields as described in "Creating Group Entries by Using the Oracle Internet Directory Self-Service Console" on page 31-20.
5. Choose **Submit**.

Deleting Group Entries by Using the Oracle Internet Directory Self-Service Console

To delete group entries:

1. Select the **Directory** tab, and perform a search for the group whose entry you want to delete.
2. From the search results, select the group whose entry you want to delete.
3. Choose **Delete**.

Assigning Privileges to Groups by Using the Oracle Internet Directory Self-Service Console

You can privilege a group to do one or more of the following:

- Create, edit, and delete new users and groups
- Assign privileges to users and to other groups

You can also revoke privileges from a group.

To assign privileges to a group:

1. Select the **Directory** tab, then select **Groups**.
2. Search for the entry of the group to which you want to assign privileges.
3. From the search results, select the group to which you want to assign privileges.
4. Choose **Assign Privilege**. The Assign Privileges to Groups window displays a list of privileges.
5. In the Assign Privileges to Groups window, select the privileges you want to assign to this group. These are described in Table C–46 on page C-44.
6. Choose **Submit**, or, to assign privileges to another user, choose **Specify Other Group** and repeat the process.

Modifying Service Properties

You can change the display name and the network address for a service. To do this:

1. Select the **Directory** tab, then select **Services**. The Services window appears displaying a list of available services.
2. In the Services window, select the service whose properties you want to modify.
3. Choose **Edit Service**. The Edit Service window appears.
4. In the Edit Service window, enter values for the fields you want to modify.
5. Choose **Submit**.

Modifying Subscription Information for a Service Recipient

You can add or remove a user from a subscription list. You can also change a recipient's start or end date.

To modify subscription information:

1. Select the **Directory** tab, then select **Services**. The Services window appears displaying a list of available services.
2. In the Services window, select the service whose properties you want to modify.
3. Choose **Edit Subscription**. The Edit Subscription window appears.
4. Select the service recipient whose subscription information you want to modify.
5. Choose **Edit**. The Edit Service Recipient window appears.

6. In the Edit Service Recipient window, enter your modifications:
   a. In the **Service Recipient** field, give this recipient a name.
   b. In the **Start Date** field, specify the date on which the recipient can begin using the service, and, in the **End Date** field, the date on which that usage ends.

   To add users to the subscription list:
   a. Choose **Add User**. This displays the Search and Select window.
   b. In the Search and Select window, perform a search for the user you want to add to the list.
   c. From the search results, select the user you want to add, then choose **Select**. This returns you to the Add New Service recipient window. The user you just added now appears in the list.

   To remove a user from the subscription list, select the user, then choose **Remove**.
7. When you have made your changes in the Edit Service Recipient window, choose **Submit**. This returns you to the Edit Subscription window.
8. Choose **Submit**.

Managing Accounts

You can unlock, enable, or disable user accounts.

Unlocking User Accounts If a user's account has been locked for any reason—for example, they failed to change their password within the specified time limit—then you can unlock it without resetting the user password. This saves you from having to explicitly tell the user the new password. Instead, the user can simply log in by using the old password.

To unlock a user's account:

1. Select the **Directory** tab, then select **Unlock Accounts**. This displays a list of locked accounts.
2. Select the account that you want to unlock.
3. Choose **Unlock**.

Enabling User Accounts If a user's account has been temporarily suspended—that is, disabled—then you can enable it. To do this:

1. Select the **Directory** tab, then select **Account**.
2. Select **Enable Accounts**. This displays a list of disabled accounts.
3. Select the account that you want to enable.
4. Choose **Enable**.

Disabling User Accounts You can temporarily suspend—that is, disable—a user's account. To do this:

1. Select the **Directory** tab, then select **Account**.
2. Select **Disable Accounts**. This displays a list of enabled accounts.
3. Select the account that you want to disable.
4. Choose **Disable**.

Configuring Resource Type Information

You can use the Oracle Internet Directory Self-Service Console to specify information for a new resource type, and, later, to modify or delete that information.

Specifying a New Resource Type To specify a new resource type:

1. Choose the **Configuration** tab, then choose **Preference**.
2. In the **Configure Resource Type Information** section, choose **Create**. The Create Resource Type window appears.
3. In the Create Resource Type window, enter values in the appropriate fields. These are described in Table C–49 on page C-48.
4. When you have entered all of the appropriate information in the Create Resource Type window, choose **Submit**. This returns you to the Preferences window. The resource type you just specified now appears under the **Resource Type Name** column.

> **See Also:** "Resource Information" on page 2-36 for a discussion of resource type information

Configuring Default Resource Access Information

If you have a large number of users, then, instead of specifying resource access information for each user entry, you can define commonly used resources that all users automatically inherit. To do this:

1. Select the **Configuration** tab, then choose **Preferences**.
2. In the **Default Resource Access Information** section, choose **Create**. The Create Resource window appears.
3. In the **Resource Name** field, specify the name of the resource or service to be accessed by the component on your behalf.
4. From the **Resource Type** list, select the type of resource to be accessed. Default options are:
   - **OracleDB**: an Oracle9*i* Database Server
   - **ExpressPDS**: Oracle Express Pluggable Data Source
   - **JDBCPDS**: Java Database Connectivity Pluggable Data Source

   Other resource types may appear in this list as specified by the administrator.
5. Choose **Next**. The Resource Access Information window appears. Enter the appropriate information into the fields.
6. Verify that you have entered all information correctly, then choose **Submit**. This returns you to the Preferences window. The default resource access information you just created now appears in the **Resource Name** column.

> **See Also:** "Resource Information" on page 2-36 for a discussion of resource access information

Part VII

Oracle Directory Integration and Provisioning Platform

This part explains the concepts, architecture, and components of the Oracle Directory Integration and Provisioning platform, and tells you how to configure and use it to synchronize multiple directories with Oracle Internet Directory. It contains these chapters:

32

Oracle Directory Integration and Provisioning Platform Concepts and Components

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

This chapter contains these topics:

About the Oracle Directory Integration and Provisioning Platform

You can reduce administrative time and costs by integrating your applications and directories—including third-party LDAP directories—with Oracle Internet Directory. The Oracle Directory Integration and Provisioning platform, a component of Oracle Identity Management, enables you to do this. For example, you might need to do the following:

- Keep employee records in Oracle Human Resources consistent with those in Oracle Internet Directory. The Oracle Directory Integration and Provisioning platform provides this synchronization through the Oracle Directory Synchronization Service.
- Notify certain LDAP-enabled applications—such as OracleAS Portal—whenever changes are applied to Oracle Internet Directory. The Oracle Directory Integration and Provisioning platform provides this notification through the Oracle Directory Provisioning Integration Service.

Throughout the integration process, the Oracle Directory Integration and Provisioning platform ensures that the applications and other directories receive and provide the necessary information in a reliable way.

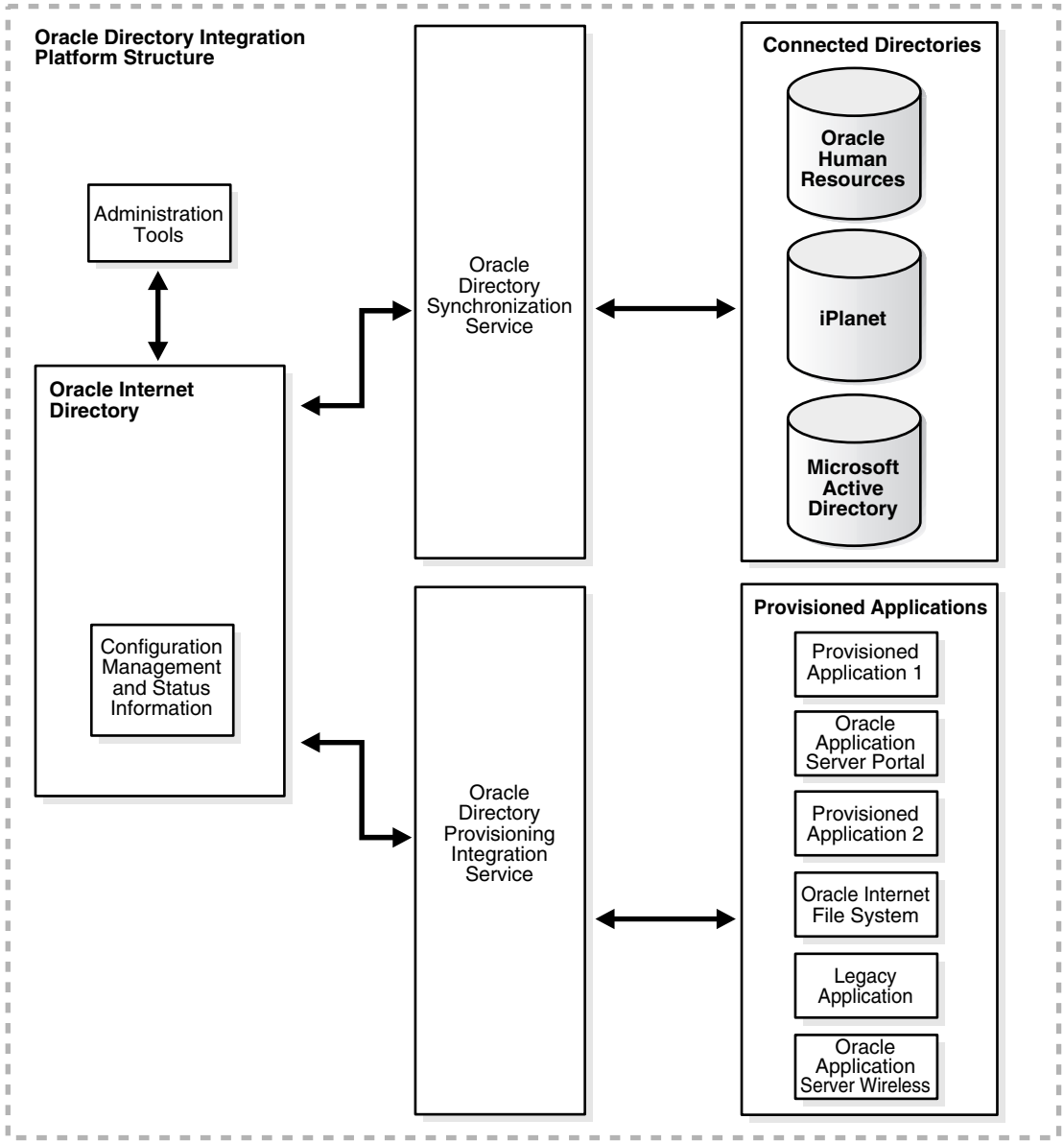
You can integrate with various directories, including Microsoft Active Directory and SunONE Directory Server. For example, in an Oracle Application Server environment, where access to Oracle components relies on data stored in Oracle Internet Directory, you can still use Microsoft Active Directory as the central enterprise directory. Users of that directory can still access Oracle components because the Oracle Directory Integration and Provisioning platform can synchronize the data in Microsoft Active Directory with that in Oracle Internet Directory.

See Also:

- Chapter 39, "Synchronization with Oracle Human Resources"
- Chapter 42, "Integration with SunONE (iPlanet) Directory Server"
- Chapter 43, "Integration with the Microsoft Windows Environment"

Figure 32–1 on page 32-3 shows a sample deployment of the Oracle Directory Integration and Provisioning platform.

Figure 32–1 Example of an Oracle Directory Integration and Provisioning Platform Environment



In the example in Figure 32–1, Oracle Internet Directory is synchronized with connected directories by way of the Oracle Directory Synchronization Service. In this example, the connected directories are Oracle Human Resources, SunONE Directory Server, and Microsoft Active Directory. Similarly, changes in Oracle Internet Directory are sent to various applications by using the Oracle Directory Provisioning Integration Service. In this example, the provisioned applications include OracleAS Portal, Oracle Content Management Software Development Kit, Oracle Application Server Wireless, an unspecified provisioned application, and a legacy application.

Synchronization, Provisioning, and the Difference Between Them

Synchronization has to do with directories rather than applications. It ensures the consistency of entries and attributes that reside in both Oracle Internet Directory and other connected directories.

Provisioning has to do with applications. It notifies them of changes to user or group entries or attributes that the application needs to track.

This section contains these topics:

- Synchronization
- Provisioning
- How Synchronization and Provisioning Differ

Synchronization

Synchronization enables you to coordinate changes among Oracle Internet Directory and connected directories. For all directories to both use and provide only the latest data, each directory must be informed of change made in the other connected directories. Synchronization ensures that any change to directory information—including, but not limited to data updated through provisioning—is kept consistent.

Whenever you decide to connect a third-party directory to Oracle Internet Directory, you create a synchronization profile for that specific directory. This profile specifies the format and content of the notifications between Oracle Internet Directory and the connected directory.

Provisioning

Provisioning enables you to ensure that an application is notified of directory changes to, for example, user or group information. Such changes can affect whether the application allows a user access to its processes and which resources can be used.

Use provisioning 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

When you install an application that you want to provision, you must create a provisioning integration profile for it by using the Provisioning Subscription Tool.

See Also: "The Provisioning Subscription Tool (oidprovtool) Syntax" on page A-127

How Synchronization and Provisioning Differ

Synchronization and provisioning have important operational differences as described in Table 32–1.

Table 32–1 Directory Synchronization and Provisioning Integration Distinctions

	Directory Synchronization	Provisioning Integration
The time for action	Application deployment time. Directory synchronization is for connected directories requiring synchronization with Oracle Internet Directory.	Application design time. Provisioning integration is for application designers developing LDAP-enabled applications.
Communication direction	Either one-way or two-way—that is, either from Oracle Internet Directory to connected directories, the reverse, or both	Two way—that is, from Oracle Internet Directory to provisioned applications, and from provisioned applications to Oracle Internet Directory
Type of data	Any data in a directory	Restricted to provisioned users and groups
Examples	Oracle Human Resources SunONE Directory Server Microsoft Active Directory	OracleAS Portal

Oracle Directory Synchronization Service

In the Oracle Directory Integration and Provisioning platform environment, the contents of connected directories are synchronized with Oracle Internet Directory through the Oracle Directory Synchronization Service.

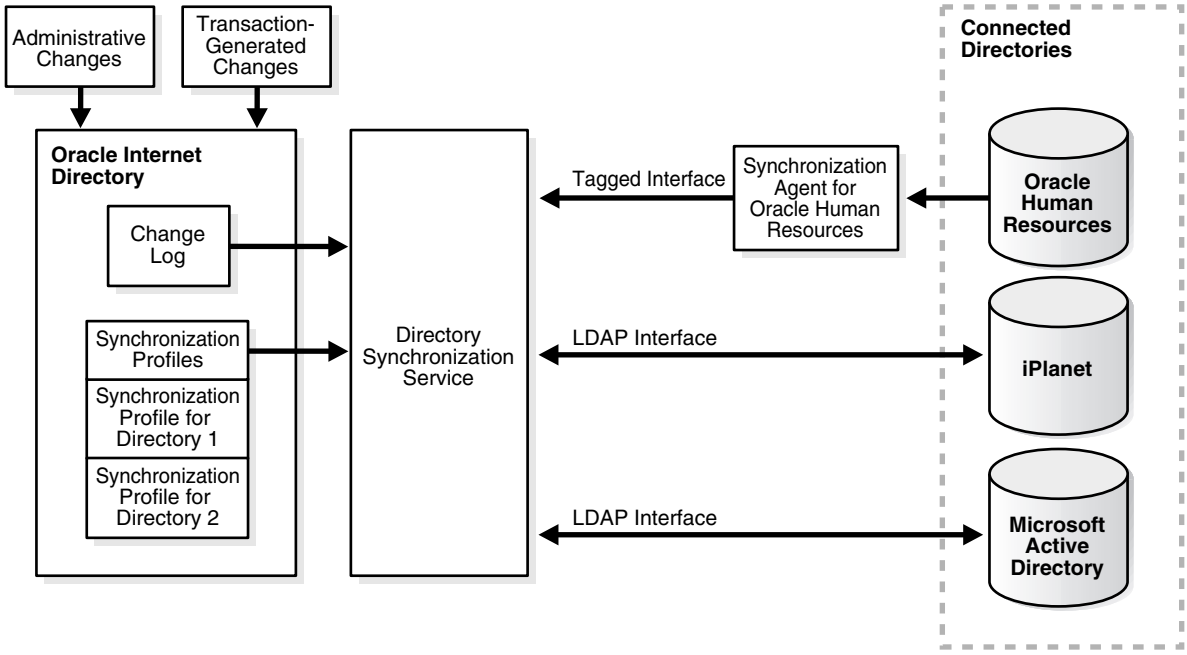
For Oracle Application Server components, Oracle Internet Directory is the central directory for all information, and all other directories are synchronized with it. This synchronization can be:

- One-way: Some connected directories only supply changes to Oracle Internet Directory and do not receive changes from it. This is the case, for example, with Oracle Human Resources, the primary repository and "source of truth" for employee information.
- Two-way: Changes in Oracle Internet Directory can be exported to connected directories, and changes in connected directories can be imported into Oracle Internet Directory.

Certain attributes can be targeted or ignored by the synchronization service. For example, the attribute for the employee badge number in Oracle Human Resources may not be of interest to Oracle Internet Directory, its connected directories or client applications. You might not want to synchronize it. On the other hand, the employee identification number may be of interest to those components, so you might want to synchronize it.

Figure 32–2 shows the interactions between components in the Oracle Directory Synchronization Service in a sample deployment.

Figure 32–2 Interactions of the Oracle Directory Synchronization Service



The central mechanism triggering all such synchronization activities is the Oracle Internet Directory change log. It adds one or more entries for every change to any connected directory, including Oracle Internet Directory. The Oracle Directory Synchronization Service:

- Monitors the change log
- Takes action whenever a change corresponds to one or more synchronization profiles
- 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 Active Directory, or SunONE Directory Server. It supplies these changes using the interface and format required by the connected directory. Synchronization through the Oracle Directory Integration and Provisioning platform connectors

ensures that Oracle Internet Directory remains up-to-date with all the information that Oracle Internet Directory clients need.

Oracle Directory Provisioning Integration Service

The Oracle Directory Provisioning Integration Service ensures that each provisioned application is notified of changes in, for example, user or group information. To do this, it relies on the information contained in a provisioning integration profile. Each provisioning profile:

- Uniquely identifies the application and organization to which it applies
- Specifies, for example, the users, groups, and operations requiring the application to be notified

The profile must be created when the application is installed, by using the Provisioning Subscription Tool.

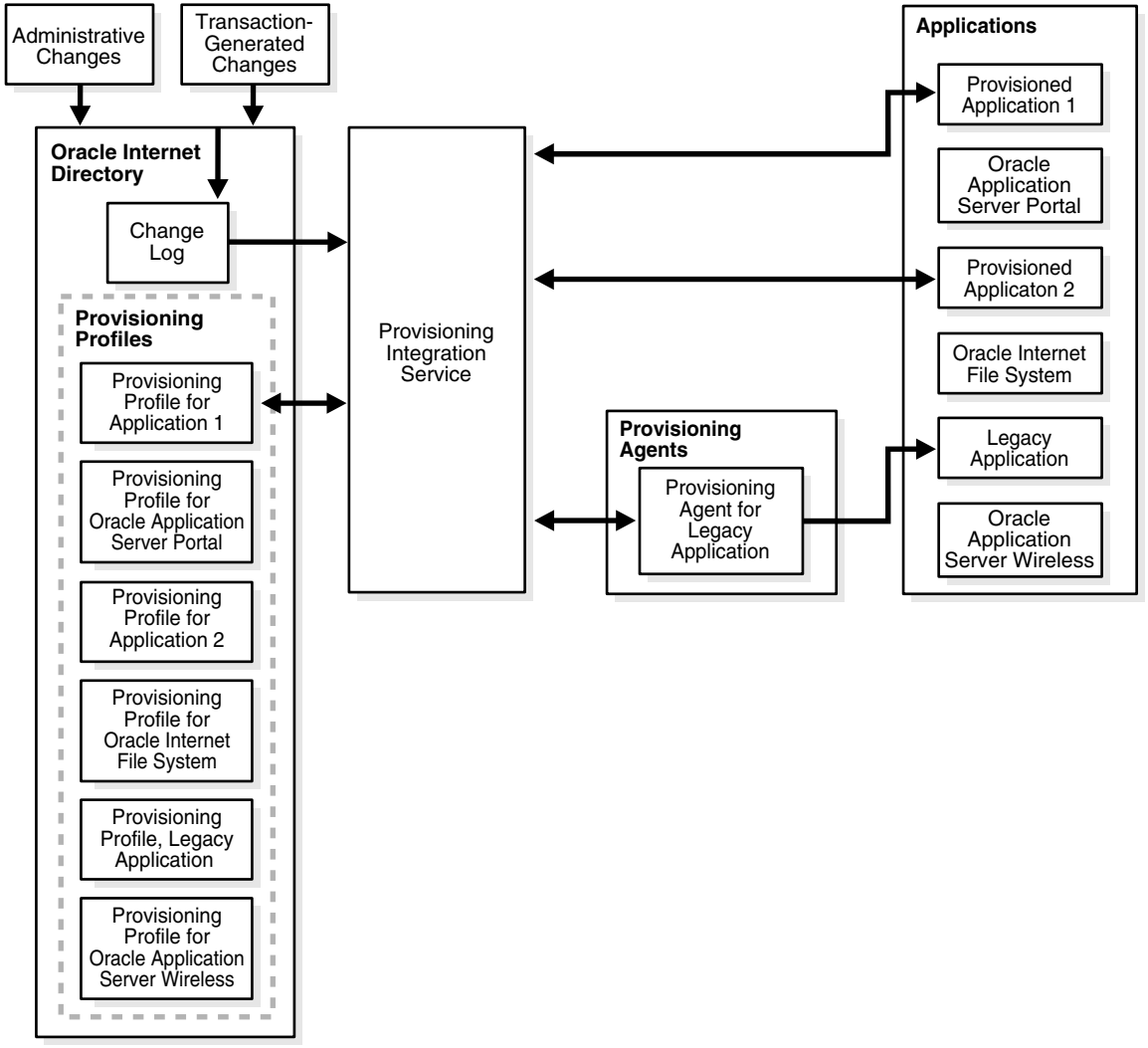
> **See Also:** "The Provisioning Subscription Tool (oidprovtool) Syntax" on page A-127 for information about the Provisioning Subscription Tool

When changes in Oracle Internet Directory match what is specified in the provisioning profile of an application, the Oracle Directory Provisioning Integration Service sends the relevant data to that application.

Note: A legacy application—that is, one that was operational before the Oracle Directory Provisioning Integration Service was installed—would not have subscribed in the usual way during installation. To enable such an application to receive provisioning information, a **provisioning agent**, in addition to the provisioning profile, must be developed. The agent must be able to translate the relevant data from Oracle Internet Directory into the exact format required by the legacy application.

Figure 32–3 shows the interactions between components in an Oracle Directory Provisioning Integration Service environment, including the special case of a provisioning agent for a legacy application.

Figure 32–3 Interactions of the Oracle Directory Provisioning Integration Service



Oracle Directory Integration and Provisioning Server

The Oracle directory integration and provisioning server is the shared server process consisting of the Oracle Directory Synchronization Service and the Oracle Directory Provisioning Integration Service. It performs these 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—Exchanging data with connected directories by using a connector
  - Error handling
- 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 35, "Oracle Directory Integration and Provisioning Server Administration"

Directory Integration Toolkit

The directory integration toolkit enables third-party vendors and developers to integrate their solutions with the Oracle Directory Integration and Provisioning 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 Oracle technology to integrate provisioning of their users and groups with Oracle Internet Directory.

The toolkit includes 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, by 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 and Provisioning platform environment. These enable you to:
  - Bulk import data from LDIF files into Oracle Internet Directory
  - Bulk export data from Oracle Internet Directory into LDIF files
- Interfaces to subscribe to user and group provisioning events—that is, changes—in Oracle Internet Directory
- Interfaces to consume changes sent by the Oracle Directory Provisioning Integration Service

Administration and Monitoring Tools

This section describes the tools you can use to administer Oracle Directory Integration and Provisioning platform. It contains these topics:

- Oracle Directory Manager
- OID Control and OID Monitor
- Directory Integration and Provisioning Assistant
- Oracle Enterprise Manager

Oracle Directory Manager

Oracle Directory Manager, a Java-based graphical user interface tool, enables you to administer the Oracle Directory Integration and Provisioning platform by:

- Creating, modifying, and deleting directory integration profiles for synchronization
- Monitoring the synchronization of directory integration profiles for synchronization
- Monitoring the status of all Oracle directory integration server instances

See Also:

- Chapter 4, "Directory Administration Tools"
- Chapter 35, "Oracle Directory Integration and Provisioning Server Administration"

OID Control and OID Monitor

OID Control and OID Monitor enable you to start, stop, and monitor the Oracle directory integration and provisioning server.

In Oracle Internet Directory, you can use OID Control and OID Monitor to control the directory integration and provisioning server in the *ORACLE_HOME* where either the Oracle directory server or Oracle directory integration server are installed.

If the 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:

- "The OID Control Utility (oidctl) Syntax" on page A-6
- "The OID Monitor (oidmon) Syntax" on page A-4
- Chapter 35, "Oracle Directory Integration and Provisioning Server Administration"

Directory Integration and Provisioning Assistant

The Directory Integration and Provisioning Assistant enables you to create, modify, and delete directory synchronization profiles and provisioning integration profiles in the Oracle Directory Integration and Provisioning platform. It also provides a bootstrap command for making Oracle Internet Directory and the connected directory contain the same data prior to exchanging information.

See Also: "The Directory Integration and Provisioning Assistant" on page A-107

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 to aid you in scheduling, monitoring, and administering your heterogeneous environment.

See Also:

- "Monitoring Oracle Internet Directory Servers" on page 10-17 for information about using Oracle Enterprise Manager to monitor Oracle Internet Directory processes
- Oracle Enterprise Manager online help

Example: A Deployment of the Oracle Directory Integration and Provisioning Platform

This section describes a deployment in which the Oracle Directory Integration and Provisioning platform integrates various applications in the MyCompany enterprise.

This section contains these topics:

Components in the MyCompany Enterprise

This hypothetical enterprise has the following components:

- Oracle Human Resources, in which all employees and contractors are managed
- An SunONE Directory Server, which is being used by certain applications
- An installation of OracleAS Portal, which is used as the intranet portal for all employees
- An installation of Oracle Content Management Software Development Kit, which is used as a document repository for all corporate documents

Requirements of the MyCompany Enterprise

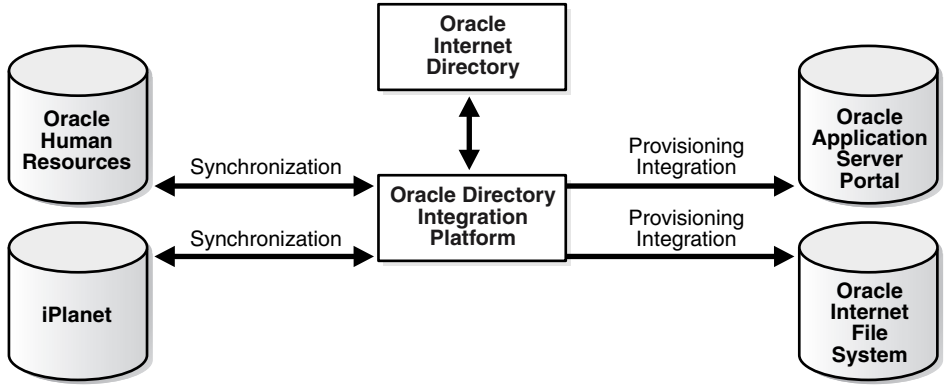
The MyCompany enterprise requires that:

- All employees and contractors are created in Oracle Human Resources. Once created, all applications in the enterprise must share this information through Oracle Internet Directory.
- All applications in the enterprise, including single sign-on services, can honor any employee created in Oracle Human Resources
- All applications interested in changes to user properties are notified when such changes occur
- A user's access rights are revoked when the user is terminated in Oracle Human Resources

Overall Deployment in the MyCompany Enterprise

Figure 32–4 illustrates the various components and their relationships to each other.

Figure 32–4 Example of Oracle Directory Integration and Provisioning Platform in the MyCompany Deployment



In the example in Figure 32–4:

- Oracle Internet Directory is the central user repository for all enterprise applications.

- Oracle Human Resources is the source of truth for all user-related information. It is synchronized with Oracle Internet Directory by using the Oracle Directory Synchronization Service.
- SunONE Directory Server, which is already deployed in the enterprise, is synchronized with Oracle Internet Directory by using the Oracle Directory Synchronization Service
- OracleAS Portal is notified of changes in Oracle Internet Directory by using the Oracle Directory Provisioning Integration Service
- Oracle Content Management Software Development Kit is notified of changes in Oracle Internet Directory by using the Oracle Directory Provisioning Integration Service.

User Creation and Provisioning in the MyCompany Enterprise

In this example, the MyCompany enterprise requires that all users be created in Oracle Human Resources. The Oracle Directory Integration and Provisioning platform must propagate new user records to all other repositories in the enterprise.

Figure 32–5 shows how the Oracle Directory Integration and Provisioning platform performs this task.

Figure 32–5 User Creation and Provisioning

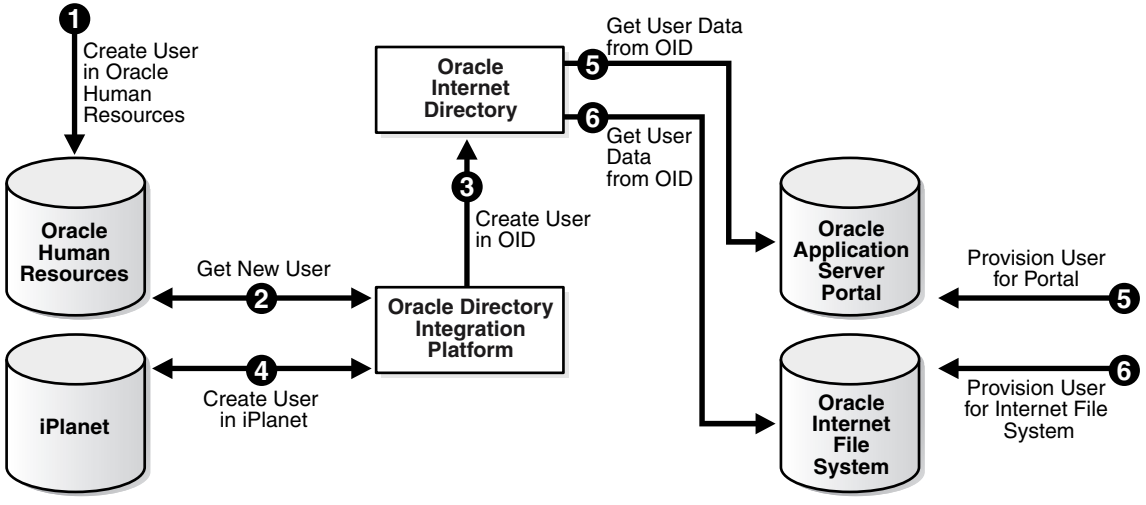


Figure 32–5 shows the creation of a new user in Oracle Human Resources, which, in turn, causes an entry for that user to be created in Oracle Internet Directory and the SunONE Directory Server. It also shows the process of provisioning the user to access two applications in the enterprise: OracleAS Portal and Oracle Content Management Software Development Kit. User creation and provisioning occur in the following manner:

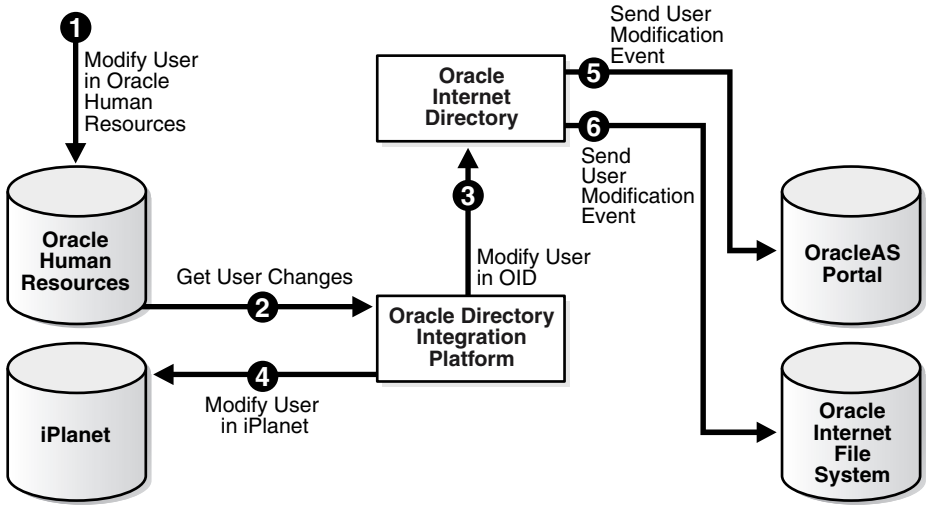
1. The Oracle Human Resources administrator creates the user in the Oracle Human Resources database.
2. The Oracle Directory Integration and Provisioning platform, through the Oracle Directory Synchronization Service, detects the new-user creation.
3. The Oracle Directory Integration and Provisioning platform, through the Oracle Directory Synchronization Service creates the entry for the user in Oracle Internet Directory.
4. The Oracle Directory Integration and Provisioning platform, through the Oracle Directory Synchronization Service, creates an entry in the SunONE Directory Server.
5. Because the user entry is available in Oracle Internet Directory, the OracleAS Portal administrator can now provision the user to use the services of OracleAS Portal. During this task, the OracleAS Portal software automatically retrieves the user details from Oracle Internet Directory.
6. The Oracle Content Management Software Development Kit administrator also provisions the user to use Oracle Content Management Software Development Kit services by using a similar process.

Note that the Oracle Directory Integration and Provisioning platform does not directly notify OracleAS Portal or Oracle Content Management Software Development Kit 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 and 6.

Modification of User Properties in the MyCompany Enterprise

In this example, the MyCompany enterprise requires that any modification to user properties must be communicated to all components interested in such changes. Figure 32–6 illustrates the actions that Oracle Directory Integration and Provisioning platform takes to meet this requirement.

Figure 32–6 Modification of User Properties



The process is as follows:

1. The user is first modified in Oracle Human Resources.
2. The Oracle Directory Integration and Provisioning platform retrieves these changes through the Oracle Directory Synchronization Service.
3. The Oracle Directory Integration and Provisioning platform makes the corresponding user modification in Oracle Internet Directory.
4. The Oracle Directory Synchronization Service modifies the user in the SunONE Directory Server.
5. The Oracle Directory Integration and Provisioning platform, through the Oracle Directory Provisioning Integration Service, notifies OracleAS Portal about the change in user properties.
6. The Oracle Directory Integration and Provisioning platform, through the Oracle Directory Provisioning Integration Service, notifies Oracle Content Management Software Development Kit about the same change in user properties.

Deletion of Users in the MyCompany Enterprise

In this example, the MyCompany enterprise requires that a user being deleted or terminated in Oracle Human Resources be automatically denied access to all enterprise resources that are based on the directory service.

Figure 32–7 shows the flow of events during the deletion of users:

Figure 32–7 Deletion of Users from the Corporate Human Resources

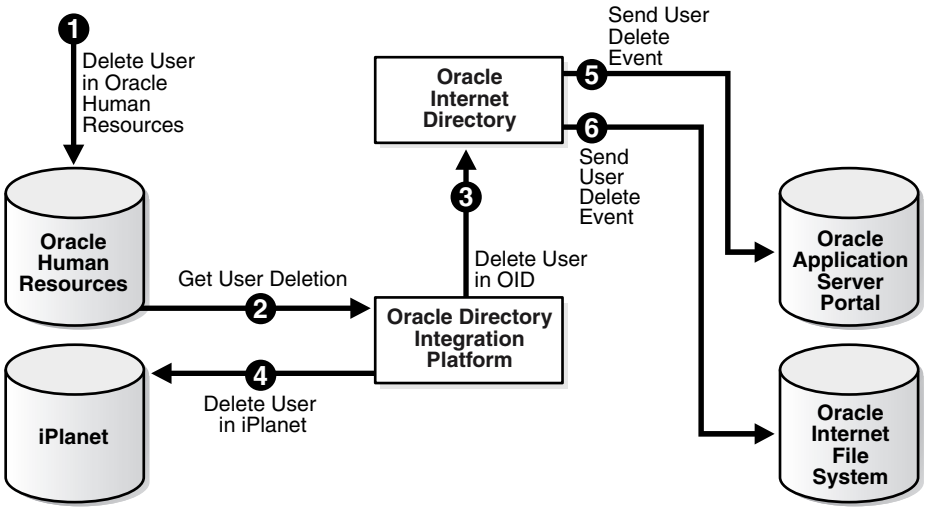


Figure 32–7 shows the process by which Oracle Directory Integration and Provisioning platform communicates the deletion of users to all systems in the enterprise. The process is as follows:

1. The user is first deleted in the Oracle Human Resources.
2. The Oracle Directory Integration and Provisioning platform retrieves these changes through the Oracle Directory Synchronization Service.
3. The Oracle Directory Integration and Provisioning platform, through the Oracle Directory Synchronization Service, makes the corresponding user deletion in Oracle Internet Directory.
4. The Oracle Directory Integration and Provisioning platform, through the Oracle Directory Synchronization Service, deletes the users in the SunONE Directory Server.

5. The Oracle Directory Integration and Provisioning platform, through the Oracle Directory Provisioning Integration Service, notifies OracleAS Portal about the deletion of the user.
6. The Oracle Directory Integration and Provisioning platform, through the Oracle Directory Provisioning Integration Service, notifies Oracle Content Management Software Development Kit about the deletion of the user.

Once all of the steps are completed, a deleted user in Oracle Human Resources can no longer access OracleAS Portal or Oracle Content Management Software Development Kit.

33

Oracle Directory Synchronization Service

This chapter discusses the synchronization profiles and connectors that link Oracle Internet Directory and connected directories. It contains these topics:

- About Connectors and Directory Integration Profiles
- Managing Synchronization Profiles
- Troubleshooting Synchronization in the Oracle Directory Integration and Provisioning Platform

See Also:

- Chapter 32, "Oracle Directory Integration and Provisioning Platform Concepts and Components" for a conceptual discussion of the Oracle Directory Integration and Provisioning platform
- "Oracle Directory Provisioning Integration Service" on page 32-8 for a discussion of the second type of integration profile, called a provisioning integration profile, which identifies the data and methods for notifying an application of changes in user or group data

About Connectors and Directory Integration Profiles

This section contains these topics:

Connectors for Directory Synchronization

To synchronize between Oracle Internet Directory and a connected directory, the Oracle Directory Integration and Provisioning platform relies on a prepackaged connectivity solution called a connector. Minimally, this connector consists of a **directory integration profile** containing all the configuration information required for synchronization.

Using Connectors with Supported Interfaces

When synchronizing between Oracle Internet Directory and a connected directory, the Oracle Directory Integration and Provisioning platform uses one of these interfaces: DB, LDAP, tagged, or LDIF. If the connected directory uses one of these interfaces, then the connector requires only a directory integration profile for synchronization to occur. For example, the SunONE connector provided with Oracle Internet Directory uses the LDAP interface to read the changes from the SunONE Directory Server. The changes are in the format specific to SunONE Directory Server and can be determined by doing an ldapsearch in the SunONE Directory Server.

Using Connectors Without Supported Interfaces

If a connected directory cannot use one of the interfaces supported by the Oracle Directory Integration and Provisioning platform, then, in addition to the directory integration profile, it requires an agent. The agent transforms the data from one of the formats supported by the Oracle Directory Integration and Provisioning

platform into one supported by the connected directory. An example is the Oracle Human Resources connector. It has both a prepackaged integration profile and an Oracle Human Resources agent. To communicate with Oracle Internet Directory, the agent uses the tagged file format supported by the Oracle Directory Integration and Provisioning platform. To communicate with the Oracle Human Resources system, it uses SQL (through an OCI interface).

Synchronization Scenarios

Depending on where the changes are made, synchronization can occur:

- From a connected directory to Oracle Internet Directory
- From Oracle Internet Directory to a connected directory
- In both directions

Regardless of the direction in which the data flows, it is assumed that:

- During synchronization, incremental changes made on one directory are propagated to the other
- Once synchronization is complete, the information maintained on both directories is the same

Synchronizing from Oracle Internet Directory to a Connected Directory

Oracle Internet Directory maintains a change log in which it stores incremental changes made to directory objects. It stores these changes sequentially based on the change log number.

Synchronization from Oracle Internet Directory to a connected directory makes use of this change log. Consequently, when running the Oracle directory integration and provisioning server, you must start Oracle Internet Directory with the default setting in which change logging is enabled. If change logging is disabled, you can enable it by using the `-l` flag in the OID Control Utility (OIDCTL) as described in "Starting an Oracle Directory Server Instance" on page A-7.

Each time the Oracle Directory Synchronization Service processes a synchronization profile, it:

1. Retrieves the latest change log number up to which all changes have been applied
2. Checks each change log entry more recent than that number

3. Selects changes to be synchronized with the connected directory by using the filtering rules in the profile
4. Applies the mapping rules to the entry and makes the corresponding changes in the connected directory

The appropriate entries or attributes are then updated in that connected directory. If the connected directory does not use DB, LDAP, tagged, or LDIF formats directly, then the agent identified in its profile is invoked. The number of the last change successfully used is then stored in the profile.

Periodically, Oracle Internet Directory purges the change log after all profiles have used what they need, and identifies where subsequent synchronization should begin.

Synchronizing from a Connected Directory to Oracle Internet Directory

When a connected directory uses DB, LDAP, tagged, or LDIF formats directly, changes to its entries or attributes can be automatically synchronized by the Oracle Directory Synchronization Service. Otherwise, the connector has an agent in its synchronization profile, which writes the changes to a file in the LDIF or tagged format. The Oracle Directory Synchronization Service then uses this file of connected directory data to update Oracle Internet Directory.

Synchronizing with Directories with Interfaces Not Supported by Oracle Internet Directory

Some connected directories cannot receive data by using any of the interfaces supported by Oracle Internet Directory. Profiles for this type of directory contain an attribute identifying a separate program for synchronization, called an agent. The agent translates between the connected directory's unique format and a DB, LDAP, tagged, or LDIF file containing the synchronization data. The agent, as identified in the profile, is invoked by the Oracle Directory Synchronization Service.

When exporting data from Oracle Internet Directory to this type of connected directory, the Oracle 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 importing data from this type of connected directory to Oracle Internet Directory, the agent creates the necessary tagged or LDIF format file. The Oracle Directory Synchronization Service then uses this file data to update the Oracle Internet Directory.

Directory Synchronization Profiles

A directory integration profile for synchronization, called a **directory synchronization profile**, contains all the configuration information required for synchronization including:

- Direction of Synchronization

  Some connected directories only receive data from Oracle Internet Directory—that is, they participate in export operations only. Others only supply data to Oracle Internet Directory—that is, they participate in import operations only. Still others participate in both import and export operations.

  A separate profile is used for each direction—that is, one profile for information coming into Oracle Internet Directory, and another for information going from Oracle Internet Directory to connected directories.

- Type of Interface

  Some connected directories can receive data in any of the interfaces built into the Oracle Directory Integration and Provisioning platform. These interfaces include LDAP, tagged, DB (for read-only), and LDIF. For these connected directories, the Oracle Directory Synchronization Service performs the synchronization itself directly, using the information stored in the profile.

- 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 to another set of attributes.

  Mapping rules govern the conversion of attributes between a connected directory and Oracle Internet Directory. Each connector stores a set of these rules in the `orclodipAttributeMappingRules` attribute of its synchronization profile. The Oracle directory integration and provisioning server uses these rules to map attributes as needed when exporting from the directory and interpreting data imported from a connected directory or file. When the Oracle directory integration and provisioning server imports changes into 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.

- Connection details of the connected directory

  These details include such information about the connected directory as host, port, mode of connection—that is, either SSL or non-SSL—and the connected directory credentials.

- Other Information

  Although the synchronization profile stores most of the information needed by a connector to synchronize Oracle Internet Directory with connected directories, some connectors may need more. This is because 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 and Provisioning platform enables you to store it in the synchronization profile as an attribute called `orclODIPAgentConfigInfo`. Its use is optional—that is, if a connector does not require such information, then simply leave this attribute empty. If such information would be useful, you can load it into this attribute by using either the Directory Integration and Provisioning Assistant or 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, the connected directory, or both. Oracle Internet Directory and Oracle directory integration and provisioning server do not modify this information. When the connector is invoked, the Oracle directory integration and provisioning server simply provides it with the information in this attribute as a temporary file.

  See Also:

  - Table B–20 on page B-18 for a list and descriptions of the attributes in a directory integration profile
  - The Directory Integration and Provisioning Assistant on page A-107 for instructions on using the Directory Integration and Provisioning Assistant
  - The ldapUploadAgentFile.sh Tool Syntax on page A-120 for instructions on using the `ldapuploadagentfile.sh` script

Registration of Connectors into the Oracle Directory Integration and Provisioning Platform

Before deploying a connector, you register it in Oracle Internet Directory. This registration involves creating a directory synchronization profile, which is stored as an entry in the directory. The attributes of this profile are listed and described in Table B–20 on page B-18.

To create the profile, you can use either Oracle Directory Manager or the Directory Integration and Provisioning Assistant as described in subsequent sections of this chapter. If you use the Directory Integration and Provisioning Assistant, then you do not need to perform a separate operation to upload the mapping and configuration files.

Most of the information needed to synchronize the data with the connected directory—such as account name, password, host name, port number—is stored in the synchronization profile. However, if the connector execution requires any additional information, it can be stored in the `orclOdipAgentConfigInfo` attribute of the synchronization profile entry as described in the previous section, "Directory Synchronization Profiles" on page 33-5.

Attributes in a synchronization profile entry belong to the object class `orclodiProfile`. The only exception is the `orclodiplastappliedchangenumber` attribute, which belongs to the object class `orclchangesubscriber`.

The Object Identifier prefix `2.16.840.1.113894.7` is assigned to platform-related classes and attributes.

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`.

Format of the Mapping Rules Attribute

The mapping rules attribute enables you to specify how to convert entries from one directory to another. You can specify domain-level mapping and attribute-level mapping. This attribute is assumed to be in the format of a file as described in this section.

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 `###`. The fields within each rule are delimited by a colon (:).

```
DomainRules
srcDomainName1: [dstDomainName1]: [DomainMappingRule1]
srcDomainName2: [dstDomainName2]: [DomainMappingRule2]
AttributeRules
srcAttrName1:[ReqAttrSeq]:[SrcAttrType]:[SrcObjectClass]:[dstAttrName1]:[DstAttr
Type]:[DstObjectClass]:[AttrMappingRule1]
srcAttrName2:[ReqAttrSeq]:[SrcAttrType]:[SrcObjectClass]:[dstAttrName2]:[DstAttr
Type]:[DstObjectClass]:[AttrMappingRule2]
###
```

where the expansion of each *srcAttrName1* and *srcAttrName2* would be a single, unwrapped long line.

Domain Level Mapping

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 33–1.

Table 33–1 DomainRule Components

Component Name	Description
`SrcDomainName`	Name of the domain or 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`. Because "import" and "export" always refer to Oracle Internet Directory, a combination of `NONLDAP:NONLDAP` is not allowed.

Table 33–1 (Cont.) DomainRule Components

Component Name	Description
`DomainMappingRule`	This field is meaningful only when importing to Oracle Internet Directory, or when exporting to an LDIF file or another external LDAP-compliant directory. This rule is used to construct the destination DN from the source domain name, from the attribute given in `AttributeRules`, or both. 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 it is not specified, then the source domain and destination domain names are considered to be the same.

Attribute Level Mapping

The attribute rule specifications appear after a line containing only the keyword `AttributeRules`. Each attribute rule is represented with the components, separated by colons, and described in Table 33–2. The attribute rule specifications end with a line containing only the characters ###.

Table 33–2 Components in Attribute Rules

Component Name	Description
`SrcAttrName`	For LDAP-compliant directory repositories, this parameter refers to the name of the attribute to be translated. For Oracle9*i* Database Server repositories, it refers to the ColumnName in the table specified by the `SrcClassName`. For other repositories this parameter can be appropriately interpreted.

Table 33–2 (Cont.) Components in Attribute Rules

Component Name	Description
ReqAttrSeq	Indicator of whether the source attribute must always be passed to the destination. When entries are synchronized between Oracle Internet Directory and the connected directory, some attributes need to be used as synchronization keys. This field indicates whether the specified attribute is being used as a key. If so, regardless of whether the attribute has changed or not, the value of the attribute is always extracted from the source. A nonzero 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—for example, integer, string, binary—that validates the mapping rules.
SrcObjectClass	If the source of the shared attribute is an LDAP-compliant directory, then this parameter names the object class to which the attribute belongs. If the source of the shared attribute is an Oracle9*i* Database Server repository, then this parameter refers to the table name and is mandatory. For other repositories, this parameter may be ignored.
DstAttrName	Optional attribute. If it is not specified, then the SrcAttrName is assumed. For LDAP-compliant directories, this parameter refers to the name of the attribute at the destination. For Oracle9*i* Database Server 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—for example, integer, string, binary. Note that it is up to you, the administrator, to ensure the compatibility of the source and destination attribute types. The Oracle Directory Integration and Provisioning platform does not ensure this compatibility.
DstObjectClass	For LDAP-compliant directories, this parameter refers to the object class to which the attribute belongs, and is optional. For Oracle9*i* Database Server repositories, it refers to the table name, and is mandatory. For other repositories this parameter may be ignored.

Table 33–2 (Cont.) Components in Attribute Rules

Component Name	Description
`AttrMapping Rule`	Optional arithmetic expression with these operators: `+`, `|`, and these functions: `toUpper (string)`, `toLower(String)`, `trunc (string,char)`. If nothing is specified, then the source attribute value is copied as the value of the destination attribute. Literals can be specified with single quotes ('') or with double quotes ("").

In a newly created synchronization profile, mapping rules are empty. To enter mapping rules, edit a file that strictly follows the correct format.

Note: When attributes and object classes are defined in the mapping file, it is assumed that source directories contain the respective attributes and object classes defined in the schema.

If a parent container is selected for synchronization, then all its children that match the mapping rules are likewise synchronized. Child containers cannot be selectively ignored for synchronization.

How to Construct a New Mapping File

To create a new mapping file, follow these steps:

1. Identify the container(s) of interest for synchronization in the source directory.
2. Identify the destination container or containers to which the objects in the source containers should be mapped to. Be sure that the specified container already exists in the directory.
3. Determine the rule to create a DN of the entry to be created in the destination directory. In LDAP-to_LDAP, mapping is normally one-to-one. In non-LDAP-to-LDAP, a domain, DN construct rule is required. For instance in the case of synchronizing from a tagged file or Human Resources agent, the mapping rule may be of the form `uid=%,dc=mycompany,dc=com`. In this case, the `uid` attribute must be present in all the changes to be applied from Oracle Human Resources. The `uid` attribute must be specified as a required attribute, as specified in step 6.
4. Identify the objects that you want to synchronize between directories—that is, the relevant object classes in the source and destination directories. In general, objects that get synchronized between directories include users, groups,

organizational units, organizations, and other resources. Identify the actual object classes used in the directories to identify these objects.

5. Identify the properties of the various objects that you want to synchronize between directories—that is, the attributes in the LDAP context. All the attributes of an object need not be synchronized. The properties of users that you might want to synchronize are `cn`, `sn`, `uid`, `mail`.

6. Define the mapping rules. Each mapping rule has this format:

```
<srcAttrName>:<ReqdFlag>:<srcAttrType>:<SrcObjectClass>:
<dstAttrName>:<dstAttrType>:<dstObjectClass>: <Mapping Rule>
```

   While defining the mapping rule, ensure the following:

   - Every required attribute has a sequence number. For example, if in step 3 the `uid` attribute is identified as required, then assign a value of `1` in place of `<ReqdFlag>`.
   - Every relevant object class has a schema definition on the destination directory.
   - Every mandatory attribute in a destination object class has a value assigned from the source. This holds good even for standard object classes also, as the different LDAP implementations may not be completely standards-compliant.

   It is not necessary to assign all attributes belonging to a source object class to a single destination object class. Different attributes of a source object class can be assigned to different attributes belonging to different destination object classes.

   If an attribute has binary values, then specify it as `binary` in the `<attrtype>` field.

Supported Attribute Mapping Rules and Examples

The attribute mapping rules supported are:

- `Concatenation (+)`: Used to concatenate two string attributes

  The mapping rule looks like:

```
Firstname,lastname:  :  :  : givenname:  : inetorgperson: firstname+lastname
```

  For example, if the `Firstname` is `John` and `LastName` is `Doe` in the source, then this rule results in the `givenname` attribute in the destination with the value `JohnDoe`.

- OR operator (|): Used to assign one of the values of the two string attributes to the destination

  The Mapping rule looks like:

  ```
  Fistname,lastname : : : :givenname: :inetorgperson: firstname | lastname
  ```

  In this example, `givenname` is assigned the value of `firstname` if it exists. If the `firstname` attribute does not exist, then `givenname` is assigned the value of `lastname`. If both the values are empty, then no value is assigned.

- `bin2b64 ( )`: Used to store a binary value of the source directory as a base64 encoded value in the destination directory. Typical usage is as follows:

  ```
  objectguid: : : :binary: :orclobjectguid: orcladuser:bin2b64(objectguid)
  ```

  This is required when you need search on the value of (`objectguid`).

- `tolower()`: Convert the String attribute value to lowercase.

  ```
  firstname: : : :givenname: :inetorgperson: tolower(firstname)
  ```

- `toupper ()`: Convert the String attribute value to uppercase.

  ```
  firstname: : : :givenname: :inetorgperson: toupper(firstname)
  ```

- `trunc(str,char)`: Truncate the string beginning from the first occurrence of the specified char

  ```
  mail : : : : uid : : inetorgperson : trunc(mail,'@')
  ```

  For example, if `mail` is `John.Doe@acme.com` in the source, then this rule results in the `uid` attribute in the destination with the value "John.Doe"

- `truncl(str, char)`: Truncate the string up to and including the first occurrence of the specified char

  ```
  mail : : : : uid : : inetorgperson : trunc(mail,'@')
  ```

  For example, if mail is `John.Doe@acme.com` in the source, then this rule results in the `uid` attribute in the destination with the value `acme.com`.

- `trunc(str1, str2)`: Truncate the string beginning with the first occurrence of the specified string

  ```
  mail : : : : uid : : inetorgperson : trunc(mail, "@")
  ```

- `dnconvert (str)`: Used for DN type attributes if domain mapping is used.

For example:

```
uniquemember : : : groupofuniquenames : uniquemember : :groupofuniquenames :
dnconvert(uniquemember)
```

In this example, if `uniquemember` in the source is `cn=test user1,cn=srcdomain`, then `uniquemember` in the destination becomes `cn=test user1,cn=dstdomain`.

if the domain mapping rules was like this.

```
DomainRules
cn=srcdomain:cn=dstdomain:
```

- Literals:

```
Userpassword: : :person: userpassword: :person: 'welcome1'
```

Example: A Mapping File for a TAGGED-File Interface

Based on the preceding discussions, here is a sample mapping file for importing user entries from the Oracle Human Resources database tables by using the tagged-file interface. This sample file is supplied during installation, at *$ORACLE_HOME*`/ldap/odi/conf/oraclehragent.map.master`.

```
DomainRules
NONLDAP:dc=myCompany,dc=com:uid=%dc=myCompany,dc=com
AttributeRules
firstname: : : :cn: :person
email : : : :cn: :person: trunc(email,'@')
email : 1 : :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 mapping file in this example contains the domain rule `NONLDAP:dc=myCompany,dc=com:cn=%,dc=myCompany,dc=com`.

- This rule implies that the source domain is NONLDAP—that is, there is no source domain.
- The destination domain (`:dc=myCompany,dc=com`) implies that all the directory entries this profile deals with are in the domain `dc=myCompany,dc=com`. Be sure that the domain exists before the start of synchronization.
- The domain mapping rule (`:uid=%,dc=myCompany,dc=com`) implies that the data from the source should refer to the entry in the directory with the DN that is constructed using this domain mapping rule. In this case, `uid` must be one of the destination attributes that 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 single-valued.
- In the case of the tagged file, the source entry does not have any object class to indicate the type of object it is synchronizing. Note that the `SrcObjectClass` field is empty.
- Every object whose destination is Oracle Internet Directory must have an object class. Specify an object class for every attribute.
- Note that `email` is specified as a required attribute in the sample mapping file. This is because the `uid` attribute is derived from the `email` attribute. Successful synchronization requires the `email` attribute to be specified in all changes specified in the tagged file as follows:

  ```
  Email : 1 : : :uid : : person : trunc(email,'@')
  ```

- In some cases, the **RDN** of the DN needs to be constructed by using the name of a multivalued attribute. For example, to construct an entry with the DN of `cn=%,l=%,dc=myCompany,dc=com`, where `cn` is a multivalued attribute, the DomainMappingRule can be of this form: `rdn,l=%,dc=myCompany,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=us,dc=myCompany,dc=com:rdn,l=%,dc=us,dc=myCompany,dc=com
AttributeRules
firstname: : :cn: :person
email : : : :cn: :person: trunc(email,'@')
email : 1: : :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: 2 : : :1: :locality
Title: : : :title: :organizationalperson
#Sex: : : :sex: :person
###
```

Mapping rules are flexible: They can include both one-to-many and many-to-one mappings.

- One-to-many

  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

  Multiple attributes in a connected directory can map to one attribute in Oracle Internet Directory. For example, suppose that the Oracle 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`. *However, in bidirectional synchronization, you cannot then map in reverse. For example, you cannot map `cn=Anne Smith` to many attributes.*

Example: Mapping Files for an LDIF Interface

A set of sample integration profiles are created as part of installation by using the Directory Integration and Provisioning Assistant. The properties file used for creating the profile is located in the directory `$ORACLE_HOME/ldap/odi/samples`.

Sample Import Mapping File

```
DomainRules
dc=mycompany.oid,dc=com:dc=mycompany.iplanet,dc=com
AttributeRules
# Mapping rules to map the domains and containers
o: : :organization: o: :organization
ou: : :organizationalUnit: ou: : organizationalUnit
dc: : :domain:dc: :domain
# Mapping Rules to map users
uid : : :person: uid: :inetOrgperson
sn: : :person:sn: :person
cn: : :person:cn: :person
mail: :inetorgperson: mail: :inetorgperson
employeenumber: :organizationalPerson: employeenumber: :organizationalperson
c: : :country:c: :country
l: : :locality: l: :locality
telephonenumber: :organizationalPerson: telephonenumber: :organizationalperson
userpassword: : :person: userpassword: :person
uid: : :person: orcldefaultProfileGroup: :orclUserV2
# Mapping Rules to map groups
cn: : :groupofuniquenames:cn: :groupofuniquenames
member: : :groupofuniquenames:member: :orclgroup
uniquemember: : :groupofuniquenames:uniquemember: :orclgroup
owner: : :groupofuniquenames:owner: :orclgroup
# userpassword: :base64:userpassword: :binary:
```

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 by using an ldapsearch command as described in "ldapsearch Syntax" on page A-39.

You cannot edit the mapping rules in Oracle Directory Manager. Instead, mapping rules are stored in a file that you upload to the directory as a value of the attribute.

To upload the mapping file, use the Directory Integration and Provisioning Assistant or the utility `ldapuploadagentfile.sh`. Once you have created and uploaded the mapping file, you can maintain a copy of it in the *$ORACLE_HOME*`/ldap/odi/conf` directory, and upload it again after any future update.

```
dipassistant mp -profile profile name odip.profile.mapfile=map file
```

See Also:

"The Directory Integration and Provisioning Assistant" on page A-107

"The ldapUploadAgentFile.sh Tool Syntax" on page A-120

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 and the object class that needs to be mapped to Oracle Internet Directory.
2. Identify the corresponding attribute name in Oracle Internet Directory and the object class to which it needs to 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.

   For instance, if the e-mail attribute of an entry in the source directory needs to be mapped to the unique identifier of the destination, then it can be:

   ```
   Email:  :  : inetorgperson: uid: : person:
   ```

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.

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.

See Also:

- "The Directory Integration and Provisioning Assistant" on page A-107 for instructions on using the Directory Integration and Provisioning Assistant
- The ldapUploadAgentFile.sh Tool Syntax on page A-120 for instructions on using the `ldapuploadagentfile.sh` script
- "Location and Naming of Files" on page 33-19 for the names of these files

---

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

Location and Naming of Files

Table 33–3 tells you where to find the various files used in the directory integration profile and during synchronization.

Table 33–3 Location and Names of Files

File	File Name
Import DataFile	`$ORACLE_HOME/ldap/odi/data/import/`*`Profile_Name`*`.dat`
Export Data File	`$ORACLE_HOME/ldap/odi/data/export/`*`Profile_Name`*`.dat`
Additional Configuration Info	`$ORACLE_HOME/ldap/odi/conf /`*`Profile_Name`*`.cfg`
Mapping Rules	`$ORACLE_HOME/ldap/odi/conf /`*`Profile_Name`*`.map`

For example, the datafile name of the Oracle Human Resources connector is `oraclehrprofile.dat`.

Managing Synchronization Profiles

This section contains these topics:

Managing Synchronization 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 directory integration profile:

1. In the navigator pane, expand in succession Oracle Internet Directory **Servers**, ***directory server instance***, **Server Management**.
2. Select **Integration Server**. The Active Processes box appears in the right pane.
3. On the toolbar, choose **Create**. The Configuration Sets dialog box appears.
4. In the Configuration Sets dialog box, choose **Create**. The Integration Profiles dialog box appears. You have two options:
   - Create an integration profile by copying an existing one

     To do this, select the Oracle Directory Integration and Provisioning platform profile you want to copy, then choose **Create Like**. The Integration Profile dialog box displays the **General** tab page.
   - Create an integration profile without copying an existing one

     To do this, choose **Create New**. The Integration Profile dialog box displays the **General** tab page.

5. Select the **General** tab page and fill in the fields. These are described in Table C–40 on page C-38.
6. Select the **Execution** tab and fill in the fields. These are described in Table C–41 on page C-39.
7. Select the **Mapping** tab and fill in the fields. These are described in Table C–42 on page C-40.
8. Select the **Status** tab and fill in the fields. These are described in Table C–43 on page C-41. Because this page shows the execution status of the connectors, most of the fields are not editable.
9. When you have entered the information, choose **OK**. This returns you to the Configuration Sets dialog box, which now lists the integration profile you just created.
10. Choose **OK** to exit the Configuration Sets dialog box. The profile you created is now registered with Oracle Internet Directory.

Deregistering a Profile by Using Oracle Directory Manager

To deregister a profile:

1. In the navigator pane, expand in succession Oracle Internet Directory **Servers**, ***directory server instance***, **Server Management**, **Directory Integration Server**.
2. Select the configuration set from which to delete the profile. The **Integration Profiles** tab page appears in the right pane.
3. In the **Integration Profiles** tab page, select the profile you want to deregister.
4. Choose **Delete**.

Changing the Synchronization Status Attribute

During synchronization in 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 by using Oracle Directory Manager:

1. Verify that the Oracle directory integration and provisioning server recognizes the disable flag for the profile.

   In the default mode, it can take up to 2 minutes for the directory integration and provisioning server to recognize this flag. To enable it to recognize this flag

sooner, set the refresh interval to a lower value as described in Table A–5 on page A-13.

2. Disable the agent by using Oracle Directory Manager.
3. Make the attribute changes.
4. Re-enable the agent after the change.

Managing Synchronization Profiles by Using Command-Line Tools

Profiles can be registered and deregistered by using the Directory Integration and Provisioning Assistant and other command-line tools. This section tells you how to register and deregister profiles.

Registering and Deregistering a Synchronization Profile by Using the Directory Integration and Provisioning Assistant

You can both create and delete a synchronization profile by using the Directory Integration and Provisioning Assistant.

> **See Also:** "The Directory Integration and Provisioning Assistant" on page A-107

Registering a Synchronization Profile by Using ldapcreateconn.sh

You can create a synchronization profile by using the command-line tool `ldapcreateconn.sh`. This tool is in the directory `$ORACLE_HOME/ldap/admin/`.

> **See Also:** "The ldapCreateConn.sh Tool Syntax" on page A-121

Deregistering a Synchronization Profile Using ldapdeleteconn.sh

You can deregister a synchronization profile by using the command-line tool `ldapdeleteconn.sh`. This tool is in the directory `$ORACLE_HOME/ldap/admin/`.

> **See Also:** "The ldapDeleteConn.sh Tool Syntax" on page A-123

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.3.2.2-1 or later. Visit:
  `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit:
  `http://www.datafocus.com/`

Troubleshooting Synchronization in the Oracle Directory Integration and Provisioning Platform

Troubleshooting synchronization can be difficult if there are a large number of profiles running or the scheduling interval for the profile of interest is very short. In such cases, the behavior of any connector can be tested as follows:

1. If there are many profiles running, use Oracle Directory Manager to selectively disable the profile that needs troubleshooting. On the other hand, if only one profile is running, stop the Oracle directory integration and provisioning server.
2. Go to `$`*`ORACLE_HOME`*`/bin` and run the command `oditest` as follows:

   ```
   oditest sync profile_name host=host_of_Oracle_Internet_Directory port=port_
   for_Oracle_Internet_Directory binddn=bind_DN bindpass=password_for_the_bind_
   DN sslauth=0 debug=63
   ```

This creates the audit and log files in the directory `$`*`ORACLE_HOME`*`/ldap/odi/log`.

To see the behaviors of synchronization, look at the trace and audit files.

34

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*

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

Provisioning involves:

- Applications subscribing to receive changes to particular data in the directory
- The directory sending those changes to the subscribing applications

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 about only 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 a given application to be notified only when members join or leave a particular group.

When it is first installed, an application subscribes to provisioning by creating a provisioning profile in the directory. There must be a profile for each application in each identity management realm.

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 information for users, groups, and user subscriptions from any of the following:

- Oracle Delegated Administration Services
- Oracle Human Resources or other applications integrated with the Oracle Directory Integration and Provisioning 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 This method 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. Oracle Application Server Single Sign-On uses this method to enroll a user accessing an application.

Manual Enrollment In this method, an 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, after the necessary approvals are complete, to enroll the user manually.

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

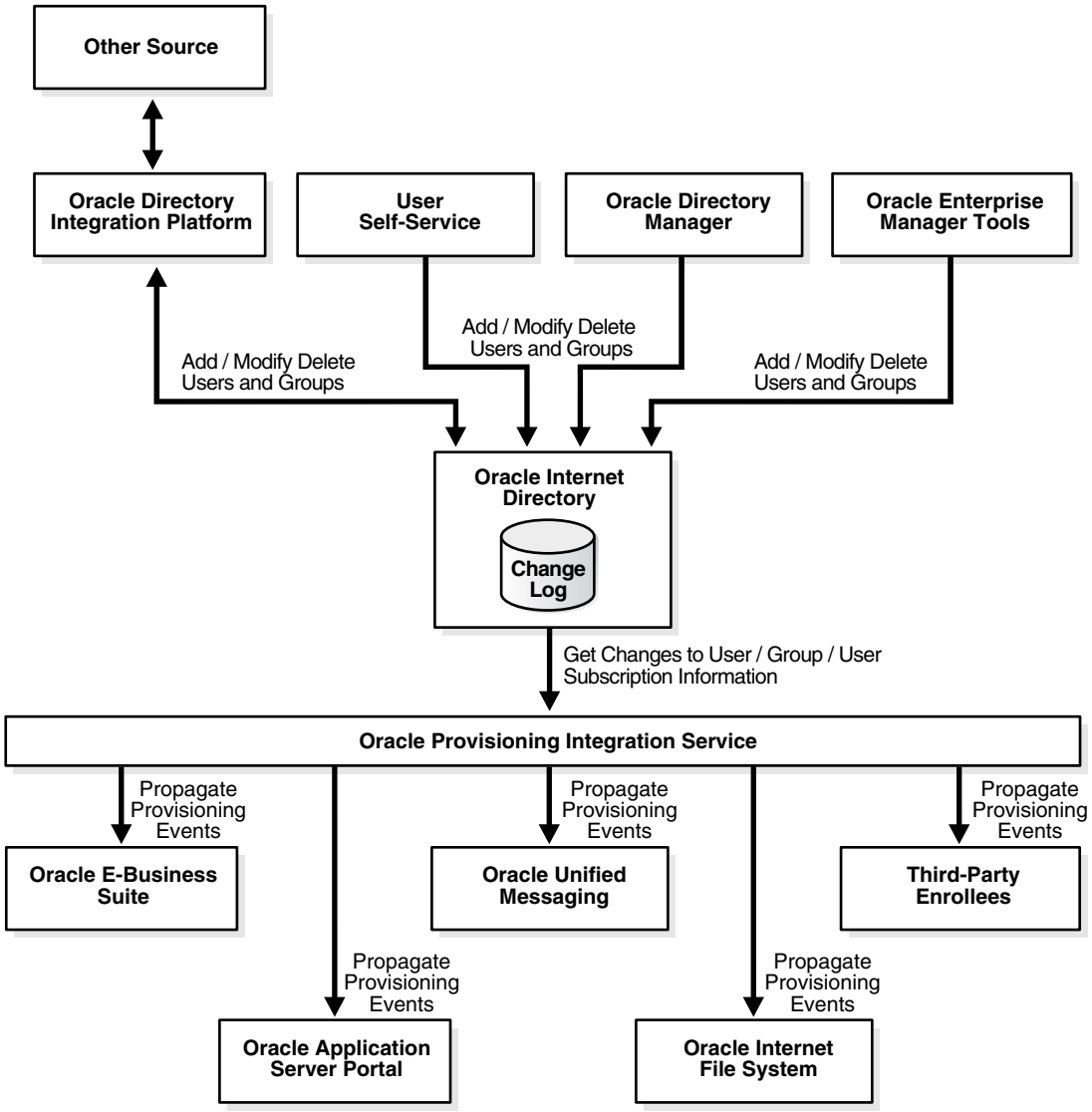
In an Oracle Directory Provisioning Integration Service environment:

- Oracle Internet Directory acts as the central repository for all information for users, groups, and user subscriptions.
- 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 relevant 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. 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 34–1 shows the relation between components in an Oracle Directory Provisioning Integration Service environment.

Figure 34–1 Typical Deployment of The Oracle Directory Provisioning Integration Service Environment



As Figure 34–1 shows:

- Oracle Internet Directory acts as the central repository for all information for users, groups, and user subscriptions
- Various components can add, modify, or delete user, group and user subscription entries in Oracle Internet Directory. These components are:
  - Oracle Directory Integration and Provisioning platform synchronizing with, for example, Oracle Human Resources or other repositories
  - The Oracle Delegated Administration Services
  - 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 information for users, groups, and user subscriptions from Oracle Internet Directory. It then sends those changes to subscribed applications. In this example, the applications are OracleAS Portal, Oracle Unified Messaging, Oracle Content Management Software Development Kit, and third-party enrollees.

How an Application Registers with the Oracle Directory Provisioning Integration Service

After the application is installed and an application identity has been created in Oracle Internet Directory, application registration with the Oracle Directory Provisioning Integration Service can occur in one of two ways:

- The application registers itself automatically during application installation by using the Provisioning Subscription Tool
- The administrator manually registers it by using the Provisioning Subscription Tool.

This registration information includes:

- 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

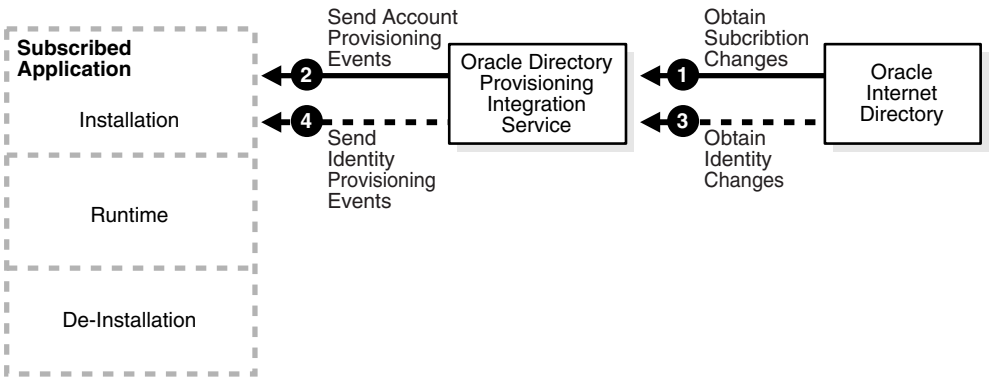
> **See Also:** Appendix A, "Syntax for LDIF and Command-Line Tools" for instructions about how to use the Provisioning Subscription Tool

How an Application Receives Provisioning Information from Oracle Internet Directory

The Oracle Directory Provisioning Integration Service monitors Oracle Internet Directory for any changes to user, group or user subscription information. It conveys these changes to applications in the form of provisioning events.

Figure 34–3 shows how an application receives the provisioning events from Oracle Internet Directory.

Figure 34–2 How an Application Receives Provisioning Information by Using the Oracle Directory Provisioning Integration Service



Provisioning information is sent from Oracle Internet Directory to an application by using the following process:

1. The Oracle Directory Provisioning Integration Service obtains from Oracle Internet Directory any changes to the subscription information for that application.

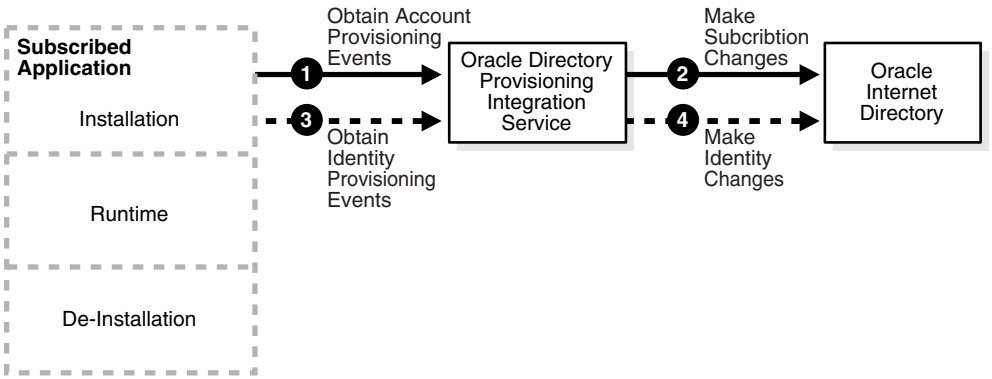
2. The Oracle Directory Provisioning Integration Service translates the subscription information to account provisioning events, which it periodically sends to the application. This information is based on application-specific database connect information.
3. The Oracle Directory Provisioning Integration Service obtains from Oracle Internet Directory any changes to the information about identities.
4. The Oracle Directory Provisioning Integration Service translates the changes to the information about identities to identity provisioning events, which it periodically sends to the application.

How Oracle Internet Directory Receives Provisioning Information from an Application

The way Oracle Internet Directory receives provisioning information from an application is the reverse of the way an application receives it from Oracle Internet Directory. That latter process was described in the previous section, "How an Application Receives Provisioning Information from Oracle Internet Directory" on page 34-7.

Figure 34–3 shows how an application sends notifications of provisioning events to Oracle Internet Directory.

Figure 34–3 How Oracle Internet Directory Receives Provisioning Information from an Application



Provisioning information is sent from an application to Oracle Internet Directory by using the following process:

1. The Oracle Directory Provisioning Integration Service obtains from the application any account provisioning events for that application.
2. The Oracle Directory Provisioning Integration Service translates the account provisioning events to subscription changes, which it periodically sends to Oracle Internet Directory.
3. The Oracle Directory Provisioning Integration Service obtains from the application any identity provisioning events for that application.
4. The Oracle Directory Provisioning Integration Service translates the identity provisioning events to identity changes, which it periodically sends to Oracle Internet Directory.

How an Application Unsubscribes from the Oracle Directory Provisioning Integration Service

You can unsubscribe an application from the Oracle Directory Provisioning Integration Service in one of two ways:

- Let the application de-install itself automatically
- Manually unsubscribe it by using the Provisioning Subscription Tool

> **See Also:** "The Provisioning Subscription Tool (oidprovtool) Syntax" on page A-127 for instructions about how to use 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 and Provisioning platform—and load user information into it.

2. Verify that the Oracle directory integration and provisioning server (odisrv) has started.
3. 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.
4. Periodically monitor the status of the provisioning event propagation for each application. You can do this by using the Oracle Enterprise Manager Application Server Control.

See Also: "Monitoring Oracle Internet Directory Servers" on page 10-17

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 and Provisioning Server" on page 35-6 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 and Provisioning platform can process it

- Modify an existing provisioning profile
- Enable or disable an existing 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 Application Server Control to monitor provisioning profiles.

See Also: the following for more details:

- "The Provisioning Subscription Tool (oidprovtool) Syntax" on page A-127
- Oracle Enterprise Manager online help

Security and the Oracle Directory Provisioning Integration Service

This section describes the principal entities in the provisioning integration process and the privileges 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. You should therefore limit the number of those who can provision applications.

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 and provisioning 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 34–1 shows the entry-level privileges granted to each entity.

Table 34–1 Entry-Level Privileges

User Category	Browse	Add	Delete	Explanation
Oracle directory integration and provisioning server	Yes	No	Yes	Oracle directory integration and provisioning servers need to: ▪ Browse all provisioning profiles ▪ Delete some rogue provisioning profiles that the applications did not bother to delete However, Oracle directory integration and provisioning 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 10*g* (9.0.4), 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 34–2 describes them.

Table 34–2 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 34–3 describes the access control for the secure attributes for the main entities operating on the provisioning profiles.

Table 34–3 Access Control for Secure Attributes

User Category	Read	Write	Search	Compare	Explanation
Oracle directory integration and provisioning servers	Yes	No	Yes	Yes	Oracle directory integration and provisioning 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 34–3 (Cont.) 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.

Table 34–4 shows the access control for all other attributes in the provisioning profiles.

Table 34–4 Access Control for All Other Attributes

User Category	Read	Write	Search	Compare
Oracle directory integration and provisioning 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 and provisioning 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 34–5 Provisioning Error Messages

Message	Reason	Remedial Action
LDAP Connection Failure	The Oracle Directory Integration and Provisioning platform failed to connect to the directory server.	Check the connection to the directory server. **See Also:** "Viewing Active Server Instance Information" on page 5-13 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:** "Manually Registering the Oracle Directory Integration and Provisioning Server" on page 35-12
Initialization Failure	Problem in connecting to the directory server using JNDI.	Look at the trace/audit file in `$ORACLE_HOME/ldap/odi/log/PROFILE_NAME.trc`
Database Connection Failure	Problem connecting to the database with the given account information. Either the database is not running or there is an authentication problem.	Look at the trace/audit file in `$ORACLE_HOME/ldap/odi/log/PROFILE_NAME.trc`
Exception while calling SQL Operation	Problem in executing the package.	Verify the package usability. Look at the trace/audit file in `$ORACLE_HOME/ldap/odi/log/PROFILE_NAME.trc`

Monitoring Provisioning Integration Profile Status Information

You can monitor certain provisioning integration profile status information from the Oracle Enterprise Manager Application Server Control.

Choose Directory Integration Server on the LDAP Main Page. This should always be green if the required packages are installed properly. This does not indicate whether the Oracle directory integration and provisioning server is running or not. To check the status of the servers, choose Integration.

On the next window, the Oracle Enterprise Manager Application Server Control displays the various running instances of the directory integration platform

servers—including those for both provisioning and synchronization. The main data displayed for provisioning integration profiles in this window are:

- Name of the subscribed application
- Name of the organization for which the subscription was made
- Status of the profile (ENABLED or DISABLED)
- Change number in Oracle Internet Directory up to which the events have been propagated to the application on behalf of this profile
- Last Execution Time
- Last Successful Execution Time of the profile.
- Errors, if any

Note that this window does not currently display the various event subscriptions for this profile.

You can also get detailed output on provisioning integration status by running from a command line `$ORACLE_HOME/bin/oidprovtool` with the operation argument `status`.

35

Oracle Directory Integration and Provisioning Server Administration

This chapter discusses the Oracle directory integration and provisioning server and how to configure and manage it. It contains these topics:

About the Oracle Directory Integration and Provisioning Server

The Oracle directory integration and provisioning server, the central component of the Oracle Directory Integration and Provisioning platform, does the following:

- Scheduling of connectors

  The directory integration and provisioning server schedules connectors for synchronizing between Oracle Internet Directory and connected directories. If there is an agent, it also schedules the execution time of the agent.

- Data import and export

  The directory integration and provisioning server imports changes into and exports changes out of Oracle Internet Directory. DB, LDAP, LDIF, and tagged interfaces are supported.

- Mapping

  The Oracle directory integration and provisioning server includes a generic facility for filtering and mapping data to and from the connected directories. The directory integration and provisioning server maps attributes when exporting data to a connected directory and when interpreting data imported from a file or directory for input to Oracle Internet Directory.

The Oracle directory integration and provisioning server performs functions for both synchronization and provisioning. You can run multiple directory integration and provisioning server instances on any host.

Operational Information about the Oracle Directory Integration and Provisioning Server

This section introduces structural and operational information about the directory integration and provisioning server and contains these topics:

- The Oracle Directory Integration and Provisioning Server and Configuration Set Entries
- Standard Sequences of Directory Integration and Provisioning Server Events
- Managing Configuration Set Entries Used by the Oracle Directory Integration and Provisioning Server

The Oracle Directory Integration and Provisioning Server and Configuration Set Entries

Each directory integration and provisioning server can execute a set of connectors either for:

- Synchronizing between Oracle Internet Directory and connected directories. The set of connectors for synchronization is provided in the configuration set number entered in the command line when starting the Oracle directory integration server.
- Provisioning users, groups, and realms for Oracle components. The set of profiles for provisioning is provided in the `groupID` argument in the command line when starting the Oracle directory integration and provisioning server.

If the configuration set number is not specified, then the directory integration and provisioning server starts in the mode for processing provisioning profiles. If the configuration set number is specified, but there are no integration profiles in the directory for the specified configuration set number, then the directory integration and provisioning 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 disabled.

If the configuration set specified in the command line does not exist in the directory, then the directory integration and provisioning server logs this information in the log file and exits. For provisioning profiles, the same behavior is followed for the `groupID` attribute, which is passed as an argument in the command line.

Whenever a connector is scheduled to do synchronization or provisioning, the directory integration and provisioning server starts up a separate thread. This thread opens an LDAP connection to the directory server to read or write entries from Oracle Internet Directory, and then closes the connection before exiting.

The directory integration and provisioning server executes three types of threads in the process, and these are described in Table 35–1:

Table 35–1 Oracle Directory Integration and Provisioning Server Threads

Thread	Description
Main thread	Daemon thread of the Oracle directory integration and provisioning server. To look for changed profiles and to refresh its cache, it starts up the scheduler and periodically sends refresh signals to it. This thread also looks for the shutdown signal from the OID Monitor (`oidmon`). This signal causes the thread to shut itself down after it sends a signal to the scheduler to shut down.

Table 35–1 (Cont.) Oracle Directory Integration and Provisioning Server Threads

Thread	Description
Scheduler thread	Scheduler for the connectors for synchronization based on their specified scheduling interval. On receipt of a refresh signal from the main thread, this thread refreshes the synchronization profiles to the latest values.
Connector thread	In a synchronization, the thread that invokes the connector executable named in the profile, and maps and filters the attributes. It is spawned by the scheduler at the specified individual scheduling intervals. Once all the changes from the source directory are propagated to the destination directory, this thread exits.

Standard Sequences of Directory Integration and Provisioning Server Events

Each instance of the Oracle directory integration and provisioning server supports either provisioning or synchronization. The directory integration and provisioning server runs as a shared server process while handling the synchronization and provisioning event propagations.

The three threads described in Table 35–1 on page 35-3 work together to create these typical process flow sequences:

- Main Thread Process Sequence
- Scheduler Thread Process Sequence
- Connector Thread Process Sequence for Synchronization
- Connector Thread Process Sequence for Provisioning

Main Thread Process Sequence

On startup, the main thread comes up. This daemon thread of the server starts the scheduler. It verifies the registration of the instance in the directory. If the instance is not registered, then it is not started up by OID Monitor. Instead, it registers itself in Oracle Internet Directory with the configuration set number and the instance number details.

The main thread periodically checks for the refresh time and signals the scheduler to refresh. It also periodically checks for the shutdown signal. On receipt of the shutdown signal, it signals the scheduler thread to shutdown.

Once the scheduler thread shuts down, the main thread unregisters and shuts down.

Scheduler Thread Process Sequence

When it is started by the main thread, the scheduler thread reads the configuration set to determine which integration profiles to schedule. It creates a list of profiles to be scheduled and schedules them based on their specified scheduling interval. While creating the list of profiles, it validates the attributes. If any of the profile attributes have invalid values, the profile is not considered for synchronization or provisioning.

When it receives the refresh signal, the scheduler thread refreshes the integration profiles. When it receives the shutdown signal, the scheduler thread waits until all the connectors complete the synchronization or provisioning event propagation. It then returns control to the main thread.

Connector Thread Process Sequence for Synchronization

A synchronization thread follows this process:

1. Establishes connection with the connected directory and Oracle Internet Directory
2. In an import operation, executes any agent execution command that may be specified in the connector
3. Opens the DB/LDAP/LDIF/Tagged file if required
4. Reads the changes from the source one at a time
5. Filters the changes if applicable
6. Maps the changes as specified by the mapping rules
7. Creates the destination change record
8. Write the changes to the destination
9. After applying all the changes, closes the thread

Connector Thread Process Sequence for Provisioning

A provisioning thread follows this process:

1. Establishes a connection with the connected directory
2. Reads the changes from the source, one at a time
3. Filters the changes if applicable
4. Identifies the change as a specific event—that is:

- USER Add/Modify/Delete
- GROUP Add/Modify/Delete

5. Creates the event notification record
6. Invokes the given package to consume the event notification

Managing the Oracle Directory Integration and Provisioning Server

This section contains these topics:

- Viewing Oracle Directory Integration and Provisioning Server Information
- Managing Configuration Set Entries Used by the Oracle Directory Integration and Provisioning Server
- Managing the SSL Certificates of Oracle Internet Directory and Connected Directories
- Starting, Stopping, and Restarting the Oracle Directory Integration and Provisioning Server
- Starting and Stopping the Oracle Directory Integration and Provisioning Server in a High Availability Scenario
- Setting the Debug Level for the Oracle Directory Integration and Provisioning Server
- Managing the Oracle Directory Integration and Provisioning Platform in a Replicated Environment
- Finding the Log Files

Note: For security reasons, Oracle Corporation recommends that you run the Oracle directory integration and provisioning server on the same host as the directory server. If you run them on different hosts, then run them by using SSL as described in Chapter 13, "Secure Sockets Layer (SSL) and the Directory".

Viewing Oracle Directory Integration and Provisioning Server Information

When the directory integration and provisioning server starts, it generates specific runtime information and stores it in the directory. This information includes:

- The instance number of the directory integration and provisioning server

- The host on which it is running
- The configuration set with which the directory integration and provisioning server was started
- The group identifier of the provisioning profile group it is running

You can view this information by using either Oracle Directory Manager or ldapsearch.

Viewing Oracle Directory Integration and Provisioning Server Runtime Information by Using Oracle Directory Manager

To view runtime information for the directory integration and provisioning server instance by using Oracle Directory Manager:

1. In the navigator pane, expand in succession **Oracle Internet Directory Servers**, *directory server instance*, **Server Management**.
2. Select **Directory Integration Server**. The Active Processes box appears in the right pane.
3. Select a directory integration and provisioning server instance, then choose **View Properties**. The Server Process dialog box displays the information.

Viewing Oracle Directory Integration and Provisioning Server Runtime Information by Using ldapsearch

To view registration information for the directory integration and provisioning 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= subregistrysubentrycn:
instance1orclodipconfigdns: orclodipagentname=HRAgent,cn=subscriber
profile,cn=changelog subscriber,cn=oracle internet directory
orcldiaconfigrefreshflag: 0
orclhostname: my_host
orclconfigsetnumber: 1
objectclass: top
objectclass: orclODISInstance
```

Managing Configuration Set Entries Used by the Oracle Directory Integration and Provisioning Server

You can create, modify, and view configuration set entries by using either Oracle Directory Manager or the appropriate command-line tools. When a connector is registered, an integration profile is created and added to the given configuration set. This configuration set entry determines the behavior of the directory integration and provisioning server.

You can control the runtime behavior of the directory integration and provisioning server by using a different configuration set entry when you start it. For example, you can start instance 1 of the directory integration and provisioning server on host H1 with configset1, and instance 2 on host H1 with configset2. The behavior of instance 1 depends on configset1, and that of instance 2 depends on configset2. Dividing the agents on host H1 between two configuration set entries distributes the load between the two directory integration and provisioning server instances. Similarly, running different configuration sets and different instances on different hosts balances the load between the servers.

Managing the SSL Certificates of Oracle Internet Directory and Connected Directories

The certificates to be used for connecting Oracle Internet Directory and connected directories are stored in a wallet by using Oracle Wallet Manager.

> **See Also:** The chapter on Oracle Wallet Manager in *Oracle Advanced Security Administrator's Guide*

The location of the wallet and the password to open it are stored in a properties file used by the Oracle Directory Integration and Provisioning platform. This file is `$ORACLE_HOME/ldap/odi/conf/odi.properties`.

A typical `odi.properties` file has the entries described in Table 35–2.

Table 35–2 Entries in the odi.properties File

Entry	Description
`RegWalletFile: odi/conf/srvWallet`	This entry indicates the location of the registration information of the Oracle Directory Integration and Provisioning platform with Oracle Internet Directory. The location of the file is in relation to the `$ORACLE_HOME/ldap` directory.

Table 35–2 (Cont.) Entries in the odi.properties File

Entry	Description
`CertWalletFile: location_of_certificate_wallet`	Location of the certificate wallet.
`CertWalletPwdFile: location_of_certificate_wallet_password_file`	Location of the certificate wallet password file that is stored in a encrypted format in a specific file. **See Also:** Chapter 13, "Secure Sockets Layer (SSL) and the Directory" "The Directory Integration and Provisioning Assistant" on page A-107

All the file locations are absolute path names. The certificate wallet file is the location of the `ewallet.p12` file.

As an example, an `odi.properties` file can look like this:

```
RegWalletFile:  /private/myhost/orahome/ldap/odi/conf
CertWalletFile:  /private/myhost/orahome/ldap/dipwallet
CertWalletPwdFile: /private/myhost/orahome/ldap/
```

In this example, the wallet file `ewallet.p12` is located in the directory `/private/myhost/orahome/ldap/dipwallet`

Starting, Stopping, and Restarting the Oracle Directory Integration and Provisioning Server

This section tells you how to start, stop, and restart the Oracle directory integration and provisioning server.

Starting the Oracle Directory Integration and Provisioning Server

The way you start the Oracle directory integration and provisioning server depends on whether your installation is a typical Oracle Internet Directory installation or an Oracle Directory Integration and Provisioning platform-only installation.

> **See Also:** "Starting the Oracle Directory Integration and Provisioning Server" on page A-11

Stopping the Oracle Directory Integration and Provisioning Server

The way you stop the directory integration and provisioning server depends on the tool that you used to start it.

> **See Also:** Stopping the Oracle Directory Integration and Provisioning Server on page A-15

Restarting the Oracle Directory Integration and Provisioning Server

If you use OID Monitor and the OID Control utility, then you can both stop and restart the directory integration and provisioning server in a single `RESTART` command. This is useful when you want to refresh the server cache immediately, rather than at the next scheduled time. When the directory integration and provisioning server restarts, it maintains the same parameters it had before it stopped.

> **See Also:** Restarting Oracle Internet Directory Server Instances on page A-16

Starting and Stopping the Oracle Directory Integration and Provisioning Server in a High Availability Scenario

You can use the Oracle directory integration and provisioning server in a simple cold failover configuration. To do this, use the Oracle Directory Integration and Provisioning Server Registration Tool (`odisrvreg`), and set the `lhost` parameter to the virtual host name `VH`.

> **See Also:** "About the Cold Failover Cluster Configuration" on page 28-2

Setting the Debug Level for the Oracle Directory Integration and Provisioning Server

You can separately control the execution of the directory integration and provisioning server and that of each connector. You can also selectively disable debugging for different connectors.

For server execution, the trace is stored in the server log. For connectors, the trace is stored in the respective trace file of each connector.

If you specify a nonzero debug level, then each trace statement in the server log file includes these trace-statement types:

- `Main`—Messages from the controller thread

- `Scheduler`—Messages from the scheduler thread

Table 35–3 Debug Types for Server Debugging

Debug Event Type	Numeric Value
Starting and stopping of different threads	1
Detail level—shows the refresh details	2
Initialization, execution, and end details of connectors	4
Details during connector execution	8
Change record of the connector	16
Mapping details of the connector	32
Execution time details of the connector	64

See Also: "Managing Synchronization Profiles" on page 33-20 for instructions on selectively debugging the threads

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 Table 35–3 on page 35-11 are logged. However, errors and exceptions are always logged.

If you do not want to debug any of the connectors, then set the debug value to `3`.

You can set the debugging levels for each connector in the profile itself.

See Also:

- Table B–20 on page B-18 for information about the debug attribute for a synchronization profile
- "The Provisioning Subscription Tool (oidprovtool) Syntax" on page A-127

Managing the Oracle Directory Integration and Provisioning Platform in a Replicated Environment

For provisioning and synchronization, the replicated directory is different from the master directory. Any profiles created in the original directory need to be recreated

in the new directory, and all configurations must be performed as in the original directory.

Finding the Log Files

Execution details and debugging information are in the log file located in the `$ORACLE_HOME/ldap/log/odisrvInstance_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`.

Any other exceptions in the server are in the file `odisrv_jvm_xxxx.log` where *xxxx* is the identifier of the process running the directory integration and provisioning server in that table.

All the profile-specific debug events are stored in the profile-specific trace file in `$ORACLE_HOME\ldap\odi\log\profile_name.trc`.

Manually Registering the Oracle Directory Integration and Provisioning Server

The Oracle directory integration and provisioning server is registered with Oracle Internet Directory during installation of the Oracle Directory Integration and Provisioning platform. This registration creates a footprint in the directory indicating the specified host as the one authorized to run the Oracle Directory Integration and Provisioning platform.

There may be times when you need to perform this registration manually on the client side, as, for example, if there is a failure during installation. You can do this by using either the Oracle directory integration and provisioning server registration tool (`odisrvreg`) or Oracle Enterprise Manager.

Manually Registering the Oracle Directory Integration and Provisioning Server by Using the Oracle Directory Integration and Provisioning Server Registration Tool

You must separately register each directory integration and provisioning server on each host by running `odisrvreg` on that host. To run this tool, you need privileges to administer a directory server.

See Also:

"The Oracle Directory Integration and Provisioning Server Registration Tool (odisrvreg)" on page A-126 for instructions on using `oidsrvreg`.

"Troubleshooting Synchronization in the Oracle Directory Integration and Provisioning Platform" on page 33-23

Manually Registering the Oracle Directory Integration and Provisioning Server by Using Oracle Enterprise Manager Application Server Control

You can use Oracle Enterprise Manager Application Server Control to configure the Oracle Directory Integration and Provisioning platform in an Oracle Identity Management infrastructure. When you do this, Application Server Control registers the Oracle directory integration and provisioning server on that infrastructure.

1. On the Application Server Control Web site, in the **Standalone Instances** section, choose the name of the Oracle Application Server instance. The corresponding screen for that instance appears.
2. Choose **Configure Components**. The Select Component screen appears.
3. Select **Oracle Directory Integration and Provisioning Platform**, then choose **Continue**. The Login screen appears.
4. Enter the user name and password of the directory super user. The default user name is `cn=orcladmin`.
5. Choose **Finish** to complete the registration.

Troubleshooting the Oracle Directory Integration and Provisioning Server

This section contains these topics:

- Troubleshooting the Oracle Directory Integration and Provisioning Server in an Infrastructure Installation
- Troubleshooting the Oracle Directory Integration and Provisioning Server in an Oracle Directory Integration and Provisioning Platform-Only Installation

Troubleshooting the Oracle Directory Integration and Provisioning Server in an Infrastructure Installation

After you start the Oracle directory integration and provisioning server, you can verify that it is running by following these steps:

1. Verify that the process is running. On Unix, you do this by using the following command:

   ```
   'ps -ef | grep odisrv'
   ```

   On Windows, check to see if the process is shown in the task bar.

2. If the Oracle directory integration and provisioning server is not running, then review the $*ORACLE_HOME*/ldap/log/oidmon.log file to see the reason for not bringing up.

3. If the log file shows any database related errors:

   a. Verify ORACLE_*SID* is set. If ORACLE_*SID* is not set, then set the value.

   b. Verify that the connect string specified in ORACLE_*SID* is specified in the $*ORACLE_HOME*/network/admin/tnsnames.ora file.

   If the log file shows such errors as invalid configset or server instance numbers, then specify valid values for those arguments.

4. If the server instance number and the configset number are correct, then look at the file $*ORACLE_HOME*/ldap/log/odisrv_xx.log where xx is the instance number of the instance started. If it indicates a registration error, then re-register the Oracle directory integration and provisioning server by using odisrvreg.

5. If no errors are indicated in step 4, then look at the file $*ORACLE_HOME*/ldap/log/odisrv_jvm_yyy.log, where yyy is the process identifier of the odisrv process that should have come up. Look for the file with the latest timestamp.

Troubleshooting the Oracle Directory Integration and Provisioning Server in an Oracle Directory Integration and Provisioning Platform-Only Installation

After you start the Oracle directory integration and provisioning server, you can verify that it is running by following these steps:

1. Verify that the process is running. On Unix, you do this by using the following command:

   ```
   'ps -ef | grep odisrv'
   ```

On Windows, check to see if the process is shown in the task bar.

2. If it is not running, then look at the file `$ORACLE_HOME/ldap/log/odisrv_xx.log` where `xx` is the instance number of the instance started. If this log file indicates a registration error, then re-register the Oracle directory integration and provisioning server by using odisrvreg.
3. If no errors are indicated in step 2, then look at the file `$ORACLE_HOME/ldap/log/odisrv_jvm_yyy.log`, where `yyy` is the process identifier of the odisrv process which should have come up. Look for the file with the latest timestamp

36

Security in the Oracle Directory Integration and Provisioning Platform

This chapter discusses the most important aspects of security in the Oracle Directory Integration and Provisioning platform. It contains these sections:

Authentication in the Oracle Directory Integration and Provisioning Platform

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 and Provisioning platform be properly authenticated before it is allowed access to the directory.

This section contains these topics:

- Secure Sockets Layer (SSL) and the Oracle Directory Integration and Provisioning Platform
- Oracle Directory Integration and Provisioning Server Authentication
- Profile Authentication

Secure Sockets Layer (SSL) and the Oracle Directory Integration and Provisioning Platform

You can deploy the Oracle Directory Integration and Provisioning platform 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 and Provisioning platform, the server is the directory server, the client is the directory integration and provisioning 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 and Provisioning platform, you must start both the Oracle directory server and Oracle directory integration and provisioning 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 and Provisioning Server Authentication

You can install and run multiple instances of the directory integration and provisioning server on various hosts. However, when you do this, beware of a malicious user either posing as the directory integration and provisioning server or using an unauthorized copy of it.

To avoid such security issues:

- Ensure that each directory integration and provisioning server is identified properly
- Ensure that, when you start a directory integration and provisioning 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 and provisioning server by using the registration tool called odisrvreg.

The registration tool creates:

- An identity entry in the directory. The directory integration and provisioning server uses this entry when it binds to the directory
- An encrypted password. It stores this password in the directory integration and provisioning 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 specified in the `odi.properties` file and it is stored in the `$ORACLE_HOME/ldap/odi/conf` directory.

When it binds to the directory, the directory integration and provisioning server uses the encrypted password in the private wallet.

Note: Ensure that the wallet is protected against unauthorized access.

See Also: "Manually Registering the Oracle Directory Integration and Provisioning Server" on page 35-12 for instructions on registering the directory integration and provisioning server

Authentication in SSL Mode

The identity of the directory server can be established by starting both Oracle Internet Directory and the directory integration and provisioning server in the SSL server authentication mode. In this case, the directory server provides its certificate to the directory integration and provisioning server, which acts as the client of Oracle Internet Directory.

The directory integration and provisioning server is authenticated by using the same mechanism used in the non-SSL mode.

You can also configure the Oracle directory integration and provisioning server to use SSL when connecting to a third-party directory. In this case, you store the connected directory certificates in the wallet as described in "Managing the SSL Certificates of Oracle Internet Directory and Connected Directories" on page 35-8.

Profile Authentication

Within Oracle Internet Directory, an integration profile represents a user with its own DN and password. The users who can access the profiles are:

- The administrator of the Oracle Directory Integration and Provisioning platform (`DIPAdmin`)
- Members of the Oracle Directory Integration and Provisioning platform administrator group (`DIPAdminGroup`)

When the directory integration and provisioning server imports data to Oracle Internet Directory based on an integration profile, it proxy-binds to the directory as that integration profile. The Oracle Directory Integration and Provisioning platform uses this mechanism to authenticate agents in both the SSL and non-SSL mode.

Access Control and Authorization and the Oracle Directory Integration and Provisioning Platform

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 and provisioning server and the agents, place appropriate access policies in the directory.

This section discusses these policies in detail. It contains these topics:

- Access Controls for the directory integration and provisioning serverOracle Directory Integration Server
- Access Controls for Agents

Access Controls for the directory integration and provisioning serverOracle Directory Integration Server

The directory integration and provisioning 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 and provisioning server to schedule synchronization actions to be carried out by various connectors.
- When the directory integration and provisioning server operates on behalf of an agent, it proxies as the agent—that is, it uses the agent credentials to bind to the directory and perform various operations. The directory integration and provisioning 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 and provisioning servers, the Oracle Directory Integration and Provisioning platform creates a group entry, called `odisgroup`, during installation. When a directory integration and provisioning server is registered, it becomes a member of this group.

You control the access rights granted to directory integration and provisioning servers by placing access control policies in the `odisgroup` entry. The default policy grants various rights to directory integration and provisioning servers for accessing the profiles. For example, the default policy enables the directory integration and provisioning server to compare user passwords for authenticating agents when it binds on their behalf. It also enables directory integration and provisioning 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:** "Security Groups" on page 14-3 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 you to control 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. Oracle Directory Integration and Provisioning platform-only installations do not create these groups and policies.

Data Integrity and the Oracle Directory Integration and Provisioning Platform

The Oracle Directory Integration and Provisioning 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 and the Oracle Directory Integration and Provisioning Platform

The Oracle Directory Integration and Provisioning 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 and provisioning server and Oracle Internet Directory, you run both components in the SSL mode.

Tools Security and the Oracle Directory Integration and Provisioning Platform

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 and provisioning server registration tool (`odisrvreg`)—Use it to register the directory integration and provisioning server in the directory.
- The Directory Integration and Provisioning Assistant when running in SSL mode
- The Provisioning Subscription Tool when running in the SSL mode

37

Bootstrapping of a Directory in the Oracle Directory Integration and Provisioning Platform

This chapter discusses directory bootstrapping—that is, migration of data between a connected directory and Oracle Internet Directory.

This chapter contains these topics:

About Directory Bootstrapping in the Oracle Directory Integration and Provisioning Platform

In the Oracle Directory Integration and Provisioning platform, bootstrapping is handled by using the Directory Integration and Provisioning Assistant with the `bootstrap` option. The command is:

```
dipassistant bootstrap
```

For information about usage of the Directory Integration and Provisioning Assistant, enter:

```
dipassistant bootstrap -help
```

The Directory Integration and Provisioning Assistant enables you to bootstrap by using either a parameter file or a completely configured integration profile. As an example of the latter method, to bootstrap from an SunONE Directory Server, configure the default integration profile that is created as part of the installation. Specifically, in that profile, enter the appropriate connected directory information and mapping rules.

This chapter discusses both approaches.

> **See Also:** "The Directory Integration and Provisioning Assistant" on page A-107

Bootstrapping by Using a Parameter File

The parameters in this file specify the source and destination data types, credentials, and the way the entries need to be mapped between Oracle Internet Directory and the connected directory. The various parameters and the default values that the Directory Integration and Provisioning Assistant assumes for them while reading the file are given in Table A–30 on page A-113.

You can bootstrap by using an LDIF file in one of these ways:

- By using the Directory Integration and Provisioning Assistant to read from the source directory
- By using directory-dependent tools to read from the source directory
- By using the Directory Integration and Provisioning Assistant to load data to Oracle Internet Directory

During installation, sample parameter files are copied to the $*ORACLE_HOME*/ldap/odi/samples/ directory. The file describes the significance of each of the parameters in bootstrapping.

When you run the tools for bootstrapping, be sure that the *ORACLE_HOME* and NLS_LANG settings are correct.

Bootstrapping can be performed between services with or without one or more intermediate files. However, for large directories, an intermediate LDIF file is required.

This section contains these topics:

- Bootstrapping Without Using an LDIF File
- Bootstrapping by Using an LDIF File

Bootstrapping Without Using an LDIF File

Oracle Corporation recommends this method for smaller directories where the entries are:

- Relatively few in number
- In a flat structure
- Not interdependent—that is, the creation of one entry does not depend on the existence of another as, for example, when the creation of a group entry depends on the existence of user member entries

To use this method:

1. Prepare the mapping file with appropriate mapping rules. The mapping file is one of the properties in the bootstrap file. Be sure that it is compatible with the mapping rules defined for synchronization.
2. Create the parameter file with the required details specifying the source as LDAP and the destination type as LDIF. A sample parameter file is available in *ORACLE_HOME*/ldap/odi/samples/ldp2ldf.properties. Make sure that binary attributes are specified as binary in the SrcAttrType field.
3. Use the Directory Integration and Provisioning Assistant bootstrap command using a configuration file in which:
   - The source is specified as an LDAP directory
   - The destination type is specified as LDIF. Dump the data to an LDIF file

   Execute the Directory Integration and Provisioning Assistant as follows:

```
Dipassistant bootstrap -cfg parameter_file
```

4. Check the `bootstrap.log` and `bootstrap.trc` files for any errors.
5. Use bulkload to upload the data to Oracle Internet Directory.
6. For continued synchronization, update the last change number:

```
dipassistant mp -profile profile_name -updcln
```

Bootstrapping by Using an LDIF File

This section describes two ways to bootstrap a directory by using an LDIF file.

Bootstrapping from an LDIF File by Using Directory-Dependent Tools to Read the Source Directory

Oracle Corporation recommends that you use this method for large directories. To use this method:

1. Download the data from the directory to an LDIF file. The tool you use depends on the directory from which you are loading the data. If you are bootstrapping from a Microsoft Active Directory, then use "ldifde" to load the data. Be sure to load all the required attributes for each entry.
2. Prepare the mapping file with appropriate mapping rules. When you want to do further synchronization, be sure that the mapping file is the same as the one used for synchronization.
3. Create the parameter file with source and destination as LDIF and other details. A sample parameter file is available in *ORACLE_HOME*`/ldap/odi/samples/ldf2ldf.properties`.
4. Use the Directory Integration and Provisioning Assistant `bootstrap` command with a parameter file in which the source is specified as `LDIF` and the destination type is specified as `LDIF`. This converts the source data and creates a new LDIF as required by Oracle Internet Directory. Execute the Directory Integration and Provisioning Assistant as follows:

   ```
   dipassistant bootstrap -cfg parameter_file
   ```

5. Check the `bootstrap.log` and `bootstrap.trc` files for any errors.
6. Use The Oracle Internet Directory bulkload tool (`bulkload.sh`) to upload the data to Oracle Internet Directory.

7. If a corresponding synchronization profile is created for further synchronization, then update the last change number:

```
dipassistant mp -profile profile_name -updcln
```

Bootstrapping from an LDIF File by Using the Directory Integration and Provisioning Assistant to Load Data to Oracle Internet Directory

To use this method:

1. Download the data from the directory to an LDIF file. The tool you use depends on the directory from which you are loading the data. If you are bootstrapping from a Microsoft Active Directory, then use "ldifde" to load the data. Be sure to load all the required attributes for each entry.
2. Prepare the mapping file with appropriate mapping rules. When you want to do further synchronization, be sure that the mapping file is the same as the one used for synchronization.
3. Create the properties file with the source specified as `LDIF` and the destination specified as `LDAP`.
4. Use the Directory Integration and Provisioning Assistant `bootstrap` command with a parameter file in which the source is specified as the LDIF file, the destination type is specified as LDAP, and the destination specified as Oracle Internet Directory. This converts the source data and creates entries in Oracle Internet Directory as required. A sample properties file, `ldf2ldp.properties`, is available in `$ORACLE_HOME/ldap/odi/samples`.
5. Check the `bootstrap.log` and `bootstrap.trc` files for any errors.
6. If a corresponding synchronization profile is created for further synchronization, then update the last change number:

```
dipassistant mp -profile profile_name -updcln
```

Bootstrapping Directly by Using the Default Integration Profile

Bootstrapping relies on an existing integration profile configured for synchronization. The configuration details are used to connect to the third-party directory.

While using this method, put the source directory in read-only mode.

If the profile is an IMPORT profile, then footprints of the required objects in the connected directory are created in Oracle Internet Directory. If the profile is an

EXPORT profile, then footprints of the required objects from Oracle Internet Directory are created in the connected directory.

While creating these entries, the domain-level and object-level mappings as specified in the integration profile are used. If there is a failure in uploading the entries, then the information is logged in `$ORACLE_HOME/ldap/odi/log/bootstrap.log`. The trace information is written to the file `$ORACLE_HOME/ldap/odi/log/bootstrap.trc`.

For example, for bootstrapping from SunONE Directory Server to Oracle Internet Directory, you would do the following:

1. Customize the default integration profile `IplanetImport`, which is created as part of installation by following the instructions in "Task 1: Configure the Integration Profile for the SunONE Connector" on page 42-5.
2. Enter the following command:

   ```
   dipassistant bootstrap -profile IplanetImport -dn  'cn=orcladmin' -passwd
   'welcome'
   ```

3. Check the `bootstrap.log` and `bootstrap.trc` files to be sure that the bootstrapping is successfully completed.

If you are bootstrapping by using the Directory Integration and Provisioning Assistant, then, at the end of the bootstrapping process, the assistant initializes the `lastchangenumber` attribute for further synchronization.

> **See Also:** "Limitations of the Directory Integration and Provisioning Assistant in Oracle Internet Directory 10g (9.0.4)" on page A-119

38

Synchronization with Relational Database Tables

This chapter explains how to synchronize data to Oracle Internet Directory from tables in a relational database. The synchronization can be either incremental—for example, one database table row at a time—or all the database tables at once.

Note: Before reading this chapter, be sure to familiarize yourself with the introductory chapters about the Oracle Directory Integration and Provisioning platform—specifically:

- Chapter 32, "Oracle Directory Integration and Provisioning Platform Concepts and Components"
- Chapter 33, "Oracle Directory Synchronization Service"

Also, be aware that Oracle Internet Directory 10*g* (9.0.4) does not enable exporting data from Oracle Internet Directory to a relational database.

This chapter contains these topics:

Overview: Synchronizing Oracle Internet Directory with Relational Database Tables

The process of synchronization with a database server involves executing a directory integration profile. This process has two steps:

1. Retrieving the data from the database. This involves executing a SQL `SELECT` statement that retrieves the specified data records from the database.
2. Writing the data into the directory. This involves converting the retrieved data records to LDAP attribute values and performing the LDAP operation on the directory.

Managing Synchronization Between Oracle Internet Directory and a Relational Database

This section describes the tasks you perform to synchronize an Oracle Internet Directory with a relational database. It also provides an example of creating an integration profile with the appropriate details.

This section contains these topics:

Task 1: Prepare the Additional Configuration Information File

During synchronization from a relational database to Oracle Internet Directory, the additional configuration information file governs the retrieval of data from the database. It provides the Oracle directory integration and provisioning server with the following information:

- The `SELECT` statement to execute
- Either the attribute(s) or the database column(s) to be used in incremental synchronization. Generally, this is either an attribute that contains a timestamp or a change sequence number that the next SQL statement should use to retrieve incremental data.

To configure this file, use the sample file `DBReader.cfg.master` in the *`ORACLE_HOME`*`/ldap/odi/samples/` directory, and edit it to your specifications.

Formatting the Additional Configuration Information File

It is very important to follow the correct format of this file. The various sections are divided using TAG names. Every TAG section has a list of parameters and their respective values. The general layout is as follows.

```
[TAG]
PARAMETER1: value
PARAMETER2: value

[TAG]
PARAMETER1: value
PARAMETER2: value\
VALUE continuation\
value continuation\
end of value continuation

[TAG]
PARAMETER1: value
PARAMETER2: value\
end of value continuation
```

For example, following this format, the `DBReader.cfg.master` file looks like this:

```
[DBQUERY]
SELECT: SELECT\
    EMPNO EmpNum,\
    ENAME,\
    REPLACE(EMAIL),'@ACME.COM','') UID,\
    EMAIL,\
    TELEPHONE,\
    TO_CHAR(LAST_UPDATE_DATE,'YYYYMMDDHH24MISS') Modified_Date\
FROM\
    EMPLOYEE\
WHERE\
    LAST_UPDATE_DATE>TO_DATE (:Modified_Date,'YYYYMMDDHH24MISS')\
ORDER BY\
LAST_UPDATE_DATE

[SYNC-PARAMS]
CHANGEKEY: Modified_Date
```

Note that the entire SELECT statement is put as a value in the parameter SELECT in the section represented by the TAG DBQUERY. Because it is a lengthy value, the value continuation character is put as the last character in every line until the SELECT statement ends.

The CHANGEKEY parameter value is the name of the column(s) to be used while doing incremental synchronization. The value(s) of these column(s) is always stored in the orclOdipLastAppliedChgNum attribute of the profile. Every time the SELECT statement is executed, the current value(s) of this attribute are put into the SQL statement accordingly. This ensures that the data is always retrieved incrementally.

If there are multiple column names in the CHANGEKEY—for example, column1:column2—then the value in the orclOdipLastAppliedChgNum attribute of the profile is stored as value1~value2 and so on, with value1 corresponding to column1 and value2 to column2.

Column names are retrieved into the Oracle Directory Integration and Provisioning platform as attribute value pairs and subsequently mapped into LDAP attribute values according to set mapping rules. For this reason, all columns names retrieved in the SELECT statement must be simple names rather than expressions. For example, you can have the expression REPLACE(EMAIL),'@ACME.COM','') but it retrieves the expression value as UID.

In this example, the Modified_Date is the key for incremental synchronization. Because it is a date, it must be represented in a string format.

When the profile is created, the orclOdipLastAppliedChgNum attribute must be set to some value. All changes after this date—that is, rows in the table with LAST_UPDATE_DATE greater than this value— are retrieved. For example, if the orclOdipLastAppliedChgNum attribute is set to 20000101000000, then all employee changes since January 1, 2000 are retrieved.

Because of the ORDER BY clause, all the database rows returned are in the order of LAST_UPDATE_DATE—that is, the changes retrieved and applied to the directory are in chronological order. Once the last change is retrieved and applied:

1. The orclOdipLastAppliedChgNum attribute value is set to the Modified_Date from the last row retrieved.
2. The profile is updated.

Whenever the Oracle Directory Integration and Provisioning platform executes the profile again, it uses the previously stored value.

Task 2: Prepare the Mapping File

To configure the mapping rules, follow the instructions in "Mapping Rules and Formats" on page 33-5.

Task 3: Prepare the Directory Integration Profile

You can create the directory integration profile by using either Oracle Directory Manager or the Directory Integration and Provisioning Assistant. If you use Oracle Directory Manager, then you must upload the additional configuration information file and the mapping file by using either the Directory Integration and Provisioning Assistant or the script `ldapUploadAgentFile.sh`

To configure the directory integration profile, follow the general instructions in "Registration of Connectors into the Oracle Directory Integration and Provisioning Platform" on page 33-7, but with these specific instructions in mind:

- Do not set a value for the Agent Execution Command (`orclodipAgentExeCommand`) attribute.
- Set the Interface Type (`orclodipDataInterfaceType`) attribute to `DB`.

See Also:

- "The Directory Integration and Provisioning Assistant" on page A-107
- "The ldapUploadAgentFile.sh Tool Syntax" on page A-120for instructions on using the ldapUploadAgentFile.sh script

Example: Synchronizing a Relational Database Table to Oracle Internet Directory

In this example, the following relational database table containing employee data is synchronized with Oracle Internet Directory.

Table 38–1 Employee Table

EMPNO	ENAME	LAST_UPDATE_DATE	EMAIL	TELEPHONE
98357	JOHN DOE	2-JAN-2000	JOHN.DOE@ACME.COM	435-324-3455
98360	ROGER BECK	3-JUL-2001	ROGER.BECK@ACME.COM	435-324-3600
98365	JIMMY WONG	4-MAR-2001	JIMMY.WONG@ACME.COM	435-324-2390
98370	GEORGE MICHAEL	6-FEB-2002	GEORGE.MICHAEL@ACME.COM	435-324-9232

You can find a sample profile for this example in the directory *ORACLE_HOME*/ldap/odi/samples. Also present there are the sample configuration and mapping files. In this example:

- The name of the table is Employee
- The Profile Name is TESTDBIMPORT.
- The employee number (EMPNO) is used to JOIN a database record with a directory entry. It is specified in the OID Matching Filter (orclOdipOIDMatchingFilter) attribute described in Table B–20 on page B-18.
- This table is present in the testsync/testsyncpwd schema in a database. The database is located on the host machine.acme.com, the database listener port is 1526 and the SID is iasdb. The database URL is machine.acme.com:1526:iasdb.
- Appropriate read/write permissions have been given explicitly to this profile, namely, orclodipagentname=testdbimport, cn=subscriber profile, cn=changelog subscriber, cn=oracle internet directory
- The profile is created in configuration set 1.

Task 1: Configure the Additional Configuration Information File

This example uses the same Additional Configuration Information file described earlier in "Task 1: Prepare the Additional Configuration Information File" on page 38-2.

Task 2: Configure the Mapping File

The mapping file for this example contains the following:

```
DomainRules
NONLDAP:dc=testdbsync,dc=com:uid=%,dc=testdbsync,dc=com
AttributeRules
ename: : : :cn: :person
ename : : : :sn: :person
uid : : : :uid: :inetOrgperson:
EMail: : : :mail: :inetOrgperson
Telephone: : : :telephonenumber: :inetOrgperson
empnum: : : :employeenumber: :inetOrgperson
```

This mapping file specifies the following:

- Directory entries are created as `uid=%,dc=testdbsync,dc=com`. The `%` is a placeholder for the actual value of `uid`. The `uid` must be present in the mapping rules so that it has a value after the mapping. Otherwise the DN construction fails.
- Both the `cn` and `sn` attributes are to have the same value as `ename`.
- The `uid` element must have the value of the `EMail` prefix, which is the element of the e-mail address prior to the '@' character.
- `empnum` becomes `employeenumber` in the directory entry.
- `telephone` becomes `telephone number` in the directory entry.

Task 3: Configure the Directory Integration Profile

The directory integration profile for this example contains the attribute values as described in Table 38–2 on page 38-7. A sample integration profile with these values populated and the corresponding mapping and configuration files are available in `$ORACLE_HOME/ldap/odi/samples` directory. You can create the profile by running the Directory Integration and Provisioning Assistant in the `createprofile` mode and specifying the file as the argument. Alternatively, you can create the profile by using Oracle Directory Manager.

See Also:

- "The Directory Integration and Provisioning Assistant" on page A-107
- "Registering a Profile by Using Oracle Directory Manager" on page 33-20 for instructions on creating a profile by using Oracle Directory Manager

Table 38–2 Directory Integration Profile for TESTDBIMPORT

Attribute	Value
Profile Name (`orclOdipAgentName`)	`TESTDBIMPORT`
Synchronization Mode (`orclOdipSynchronizationMode`)	`IMPORT`
Professoriats (`orclOdipAgentControl`)	`ENABLE`

Table 38–2 (Cont.) Directory Integration Profile for TESTDBIMPORT

Attribute	Value
Agent Execution Command (orclodipAgentExeCommand)	null
Additional Config Info (orclOdipAgentConfigInfo)	As shown in the preceding file. Needs to be uploaded
Connected Directory Account (orclOdipConDirAccessAccount)	testdbsync
Connected Directory Account Password (orclOdipConDirAccessPassword)	testdbsyncpwd
Connected Directory URL (orclOdipConDirURL)	machine.acme.com:1526:iasdb
Interface Type (orclodipDataInterfaceType)	DB
Mapping File:	To be uploaded from a file
OID Matching Filter (orclOdipOIDMatchingFilter)	employeenumber This means that employeenumber is used to search the directory while looking for a match. If a match is found, then the directory entry is modified. Otherwise, a new entry is created. This is necessary to ensure that the orclOdipOIDMatchingFilter attribute is unique in the database also. Once a database row is retrieved, the Oracle directory integration and provisioning server searches the directory for that employeenumber in the domain dc=testdbsync,dc=com according to the domain rules. If it gets a match, it updates that entry with the latest values of the columns in the row retrieved. If it does not get a match, it creates a new entry in the directory with all the attributes from the column values.
Last Applied Change Number (orclodipConDirLastAppliedChgNum)	20000101000000 This means that the first time the profile executes, it retrieves and synchronizes all four rows. Subsequently, it retrieves rows only when the LAST_UPDATE_DATE column in the table is updated to the time last modified.

Task 4: Upload the Additional Configuration Information File

If you used Oracle Directory Manager to create the profile, then enter this command:

```
ORACLE_HOME/ldap/odi/admin/ldapuploadagentfile.sh -name "TESTDBIMPORT" -config 1
\
-bindpass password -binddn "cn=orcladmin" -attrtype "ATTR" \
-filename full_path_name_of_the_file
```

Task 5: Upload the Mapping File

If you used Oracle Directory Manager to create the profile, then enter this command:

```
ORACLE_HOME/ldap/odi/admin/ldapuploadagentfile.sh -name "TESTDBIMPORT" -config 1
\
-bindpass password -binddn "cn=orcladmin" -attrtype "MAP" \
-filename full_path_name_of_the_file
```

The Synchronization Process

In this example, the sequence of steps in the synchronization process is:

1. The Oracle directory integration and provisioning server starts a new profile thread for the TESTDBIMPORT profile every time the value specified in the scheduling interval (`orclOdipSchedulingInterval`) attribute expires.
2. The profile thread reads the additional configuration information to get the SQL to execute, and then runs the SQL.
3. For every row retrieved from the database, the mapping rules are applied to the record and LDAP attributes are created.
4. Depending on the OID Matching Filter (`orclOdipOIDMatchingFilter`) attribute, the directory integration and provisioning server determines whether a matching entry exists in Oracle Internet Directory or not. If it exists, then it is updated. If not, then a new entry is created. After the directory operation, the last applied change number (`orclodipConDirLastAppliedChgNum`) attribute is updated.

Observations on the Example

When a row is retrieved from the database, it is in the following form:

```
EmpNum: 98357
EName: JOHN DOE
UID: JOHN.DOE
```

```
EMAIL: JOHN.DOE@ACME.COM
TELEPHONE: 435-324-3455
Modified_Date: 20000102000000
```

After the mapping is performed on this record, the output is in the following form:

```
dn: uid=john.doe,dc=testdbsync,dc=com
uid: JOHN.DOE
cn: JOHN DOE
sn: JOHN DOE
mail: JOHN.DOE@ACME.COM
employeenumber: 98357
telephonenumber: 435-324-3455
objectclass: person
objectclass: inetorgperson
```

A subtree search is made in the directory with the filter `employeenumber=98357` under the domain `dc=testdbsync,dc=com`. If the search yields an existing entry, then that entry is updated. Otherwise, a new entry is created. Because the OID Matching Filter (`orclOdipOIDMatchingFilter`) attribute is set to `employeenumber`, every database record retrieved must have that column. In this case, it is `EmpNum` as it maps to `employeenumber`.

Any other attributes in the mapping file that are not in the data retrieved by the SQL are ignored—for example, the attribute `birthday`.

After the profile thread processes all the change records from the SQL, it updates the directory with correct values for these attributes:

- Last Applied Change Number (`orclodipConDirLastAppliedChgNum`)
- Last Execution Time (`orclOdipLastExecutionTime`)
- Last Successful Execution Time (`orclOdipLastSuccessfulExecutionTime`)

39

Synchronization with Oracle Human Resources

If you use Oracle Human Resources as the source of truth for employee data in your enterprise, then you must synchronize between it and Oracle Internet Directory. 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:

> **See Also:** Oracle Internet Directory Release Notes to find out which release of Oracle Human Resources can be synchronized with this release of Oracle Internet Directory

Introduction to Synchronization with Oracle Human Resources

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. You can run it once you have configured the parameters to meet the needs of your deployment.

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. You can also set and modify mapping between column names in Oracle Human Resources and attributes in Oracle Internet Directory.

The Oracle Human Resources has an agent executable named `odihragent` that is located in the `$ORACLE_HOME/ldap/odi/bin` directory. You can manage the profile by using either the Directory Integration and Provisioning Assistant or Oracle Directory Manager.

Data that You Can Import from Oracle Human Resources

Table 39–1 lists the tables in the Oracle Human Resources schema. If you choose, you can import most of these attributes into Oracle Internet Directory.

Table 39–1 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 39–2 shows some of the fields in the Oracle Human Resources user interface. These fields appear when you add or modify employee data.

Table 39–2 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 39–2 (Cont.) 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 Between Oracle Human Resources and Oracle Internet Directory

This section contains these topics:

- Task 1: Configure a Directory Integration Profile for the Oracle Human Resources Connector
- Task 2: Configure the List of Attributes to Be Synchronized with Oracle Internet Directory
- Task 3: Configure Mapping Rules for the Oracle Human Resources Connector
- Task 4: Prepare for Synchronization from Oracle Human Resources to Oracle Internet Directory

Task 1: Configure 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. During installation, a default integration profile is created. The parameters in that default integration profile are listed and described in Table B–20 on page B-18. For some of those parameters, you must specify values specific to integration with the Human Resources Connector. The parameters specific to the Human Resources Connector are listed in Table 39–3 on page 39-5.

Table 39–3 Attributes Specific to 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`.
Synchronization Mode `ModeorclODIPSynchronizationMode`)	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 10*g* (9.0.4) supports import operations only for Oracle Human Resources.
Execution Information	

Agent Execution Command (`orclODIPAgentExeCommand`)	Connector executable name and argument list used by the directory integration and provisioning server to execute the connector.
	This attribute is mandatory and modifiable.
	The default is:
	`odihragent OracleHRAgent 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 connected directory to be used by the connector for synchronization. For the Human Resources Agent, it is a valid user identifier in the Oracle Human Resources database. **See Also:** Chapter 39, "Synchronization with Oracle Human Resources" for typical usage of passing it in the command-line

Table 39–3 (Cont.) Attributes Specific to Oracle Human Resources Connector Integration Profile

Attribute	Description
Additional Config Info (`orclODIPAgentConfigInfo`)	Any configuration information that you want the connector to store in Oracle Internet Directory. It is passed by the directory integration and provisioning server to the connector at time of connector invocation. The information is stored as an attribute and the directory integration and provisioning 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. **See Also:** "Task 2: Configure the List of Attributes to Be Synchronized with Oracle Internet Directory" on page 39-7 This attribute is mandatory for the Oracle Human Resources connector, and modifiable by editing the configuration file and uploading it again into the profile. You cannot modify this attribute by using Oracle Directory manager.
Connected Directory URL	The host and port details of the connected directory. It must be entered in this format: *`host`*:*`port`*:*`sid`*.
Interface Type (`orclODIPInterfaceType`)	The interface 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 Human Resources Profile.
Mapping Information	
Mapping Rules (`orclODIPAttributeMappingRules`)	Attribute for storing the mapping rules. Store the mapping rules in a file by using the Directory Integration and Provisioning Assistant or the `ldapuploadagentfile.sh` tool. This attribute is mandatory for Oracle Human Resources and is modifiable. **See Also:** ■ "Mapping Rules and Formats" on page 33-5 ■ "Format of the Mapping Rules Attribute" on page 33-7 ■ "The Directory Integration and Provisioning Assistant" on page A-107
Connected Directory Matching Filter (`orclODIPConDirMatchingFilter`)	This is not used in Oracle Human Resources connectivity.

Table 39–3 (Cont.) Attributes Specific to Oracle Human Resources Connector Integration Profile

Attribute	Description
OID Matching Filter (orclODIPOIDMatchingFilter)	This attribute names an LDAP filter that is used to search for a target entry in Oracle Internet Directory. The Oracle directory integration and provisioning 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	
OID Last Applied Change Number (orcllastappliedChangenumber)	This attribute, standard for all EXPORT profiles, does not apply to Oracle Human Resources synchronization.
Last Applied Change Number (orclODIPConDirLastAppliedChgNum)	This attribute, standard for all profiles, does not apply to the Oracle Human Resources synchronization.

Task 2: Configure the List of Attributes to Be Synchronized with Oracle Internet Directory

The default Oracle Human Resources profile provides a default list of attributes to be synchronized from Oracle Human Resources to Oracle Internet Directory. You can customize this list, adding attributes to it or removing attributes from it.

The default attribute list is stored in the orclodipAgentConfigInfo attribute as part of the integration profile.The configuration information is also available in the file oraclehragent.cfg.master that 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

Column	Description
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
```

Modifying 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 and provisioning server generates a configuration file with that name, using it to pass the configuration information to the Oracle Human Resources agent at run time.
2. Include an additional Oracle Human Resources attribute for synchronization by adding a record to this file. To do this, you need this information:
   - Table name in the database from which the attribute value is to be extracted. These tables are listed in Table 39–1 on page 39-2. 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 that is used to identify this attribute in the output file. This tag is 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.

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 and provisioning

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`

Configuring a SQL SELECT Statement in the Configuration File to Support Complex Selection Criteria

If the previous supporting attribute configuration is not sufficient to extract data from the Oracle Human Resources database, then the Oracle Human Resources agent also supports execution of a preconfigured SQL `SELECT` statement in the configuration file. There is a TAG to indicate this in the config file, namely, a `[SELECT]` in the configuration file.

The following example shows a sample select statement to retrieve some information from the Oracle Human Resources database. Note that only the SQL statement should be below the `[SELECT]` Tag. The `BINDVAR` Bind Variable needs to be there to retrieve incremental changes. The substitutes passes this value (the time stamp) to the Oracle Human Resources connector.

All the columns expressions retrieved in the `SELECT` statement must have column names—for example, `REPLACE(ppx.email_address),'@ORACLE.COM','')` is retrieved 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'''`.

The following is an example of a a `SELECT` statement in a configuration file.

```
[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')
```

Task 3: Configure Mapping Rules for the Oracle Human Resources Connector

Attribute mapping rules govern how the directory integration and provisioning server converts attributes between Oracle Human Resources and Oracle Internet Directory. You can customize the mapping rules you want the directory integration and provisioning server to use.

The Oracle Human Resources agent profile has a default mapping file with a set of mapping rules in the attribute `orclodipAttributeMappingRules`. This information is also stored 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.

See Also: "Mapping Rules and Formats" on page 33-5 for the contents of the `oraclehragent.map.master` and a description of the format of the mapping rules records

Task 4: Prepare for 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.

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 and provisioning 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 Release Notes for Oracle Internet Directory 10*g* (9.0.4) 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. Configure an integration profile for the Oracle Human Resources connector, as described in "Task 1: Configure a Directory Integration Profile for the Oracle Human Resources Connector" on page 39-4. 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
4. Once everything is properly set, set the Profile Status (`orclodipagentcontrol`) attribute to `ENABLE`. This indicates that the Oracle Human Resources connector is ready to run.
5. Start the Oracle directory server and the Oracle Human Resources system if they are not already running on the respective hosts.

6. When everything is ready, start the directory integration and provisioning server if it is not already running on this host.

   See Also: "Starting, Stopping, and Restarting the Oracle Directory Integration and Provisioning Server" on page 35-9 for instructions about starting and stopping the directory integration and provisioning server

The Synchronization Process

Once the Oracle Human Resources system, Oracle Internet Directory, and the directory integration and provisioning server are running and the Oracle Human Resources connector is enabled, the directory integration and provisioning 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 and provisioning server invokes the Oracle Human Resources connector.
2. The Human Resources agent extracts:
   - All the changes from the Oracle Human Resources System based on the time specified in the `orclodipLastSuccessfulExecutionTime` attribute in the integration profile
   - Only the attributes specified in the `orclodipAgentConfigInfo` attribute in the profile

   It then writes the changes into the Oracle Human Resources import file, namely `$`*`ORACLE_HOME`*`/ldap/odi/import/HR_Agent_Name.dat`.
3. After the agent completes the execution, it creates a data file that looks something like the following:

```
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
```

4. The Oracle directory integration and provisioning server imports the changes to Oracle Internet Directory by doing the following:
   - Reading each change record from the import file
   - Converting each change record into an LDAP change entry based on the rules specified in the Mapping Rules (`orclodipAttributeMappingRules`) in the integration profile.
5. After importing all the changes 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`) are updated to the current time.

   If the import operation fails, only the Last Execution Time (`orclodipLastExecutionTime`) attribute is updated, and the connector once again attempts to extract the changes from Human Resources system based on the Last Successful Execution Time (`orclodipLastSuccessfulExecutionTime`) attribute. The reason for failure is logged in the trace file in $*ORACLE_HOME*`/ldap/odi/HR_Agent_Name.trc` file.

Boostrapping Oracle Internet Directory from Oracle Human Resources

There are two ways to bootstrap Oracle Internet Directory from Oracle Human Resources:

- Use the Oracle Human Resources connector. In the integration profile, set the `orclodipLastSuccessfulExecutionTime` to a time before Oracle Human Resources was installed.
- Use external tools to migrate data from Oracle Human Resources into Oracle Internet Directory

40

Integration of Provisioning Data with the Oracle E-Business Suite

In Oracle Internet Directory 10*g* (9.0.4), you can use the Oracle Directory Provisioning Integration Service to synchronize user accounts and other user information from the Oracle E-Business Suite.

> **See Also:** Oracle E-Business Suite documentation for further details on this integration and how to administer it

41

Considerations for Integrating with Third-Party Directories

This chapter discusses the decisions you need to make before integrating with a third-party directory. Once you have made these decisions, you can configure bootstrapping and synchronization between the directories. This chapter assumes that you are familiar with:

- Chapter 19, "Deployment of Oracle Identity Management Realms"
- Chapter 30, "Oracle Delegated Administration Services"
- Chapter 31, "Oracle Internet Directory Self-Service Console"

This chapter contains these topics:

General Considerations for Integrating with a Third-Party Directory

If you are deploying Oracle Internet Directory in an enterprise that already has an LDAP directory server, then you must configure both directories to co-exist in the same environment. If your environment supports enterprise users from a database server point of view, then configure simple synchronization between the directories. However, if it uses a third-party directory as the enterprise directory and deploys an Oracle Application Server suite of applications, then, before configuring synchronization, you must configure the identity management realm. Thus, the co-existence of directories can require either of two different types of deployments:

- Simple synchronization with Oracle Internet Directory to support Enterprise User Security
- Complete integration with the Oracle Application Server infrastructure to enable all enterprise users for the various applications in the Oracle Application Server suite

This section contains these topics:

Configuring Simple Synchronization with a Third-Party Directory

To configure simple synchronization with SunONE Directory Server, see Chapter 42, "Integration with SunONE (iPlanet) Directory Server".

To configure simple synchronization with Microsoft Active Directory, see Chapter 43, "Integration with the Microsoft Windows Environment".

Configuring Complete Integration with the Oracle Application Server Infrastructure

Because all Oracle Application Server components depend on the identity management realm, complete integration with the Oracle Application Server infrastructure requires that you make some basic decisions about the container for that realm. Once you have made these decisions, you can configure bootstrapping and synchronization between the directories.

Choose Which Directory Is to Be the Central Enterprise Directory

The central enterprise directory is the source of truth for all user, group, and realm information in the enterprise. It can be either Oracle Internet Directory or a third-party directory.

This section contains these topics:

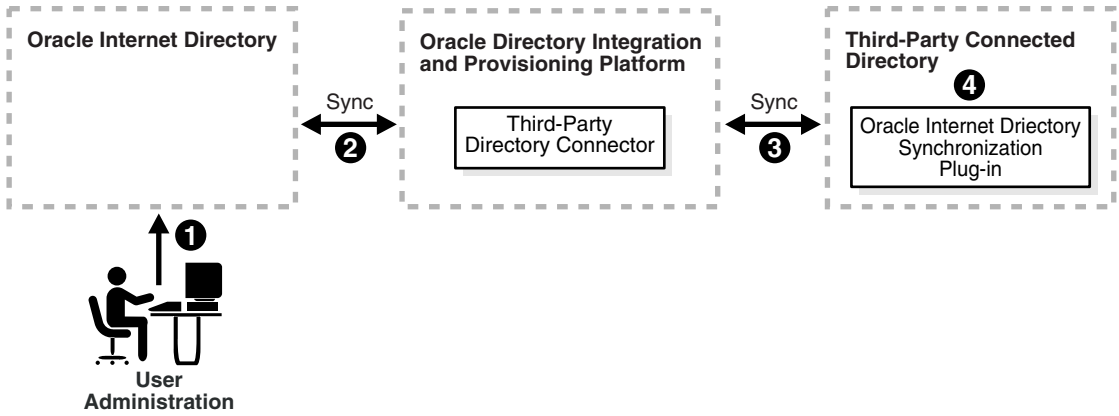
- Oracle Internet Directory as the Central Enterprise Directory
- Third-Party Directory as the Central Directory

Oracle Internet Directory as the Central Enterprise Directory

If Oracle Internet Directory is the central directory, then, once user, group, and realm objects are created, Oracle Internet Directory becomes the source of provisioning information for all Oracle components and third-party directories. The user and group objects for the entire enterprise are then provisioned in various Oracle components and third-party directories from Oracle Internet Directory.

Figure 41–1 shows a typical deployment in which Oracle Internet Directory is the central enterprise directory.

Figure 41–1 Interaction Between Components with Oracle Internet Directory as the Central Directory



As Figure 41–1 on page 41-3 shows, when Oracle Internet Directory is the central enterprise directory, typical provisioning of a user or group follows this process:

1. The user or group entry is created in Oracle Internet Directory by using the Oracle Internet Directory Self-Service Console, Oracle Directory Manager, or the command-line tools.
2. At the next scheduled interval, that entry creation event is read by the third-party directory connector in the Oracle Directory Integration and Provisioning platform.
3. Following the mapping information in the integration profile, the user or group attributes in Oracle Internet Directory are appropriately mapped to the corresponding user or group attributes as required by the schema in the third-party directory.
4. The user and group entry is created in the third-party directory.

A user entry is modified in Oracle Internet Directory, when:

- A new attribute gets added to the entry
- The value of an existing attribute is modified
- An existing attribute is deleted

When Oracle Internet Directory is the central enterprise directory, the sequence of events during modification of a user or group entry is as follows:

1. The entry is modified by using the Oracle Internet Directory Self-Service Console, Oracle Directory Manager, or the command-line tools.
2. At the next scheduled interval, that entry modification event is read by the third-party directory connector in the Oracle Directory Integration and Provisioning platform,
3. Following the mapping information in the integration profile, the attribute in Oracle Internet Directory is appropriately mapped to the corresponding attribute in the connected directory
4. The user entry is modified in the third-party directory.

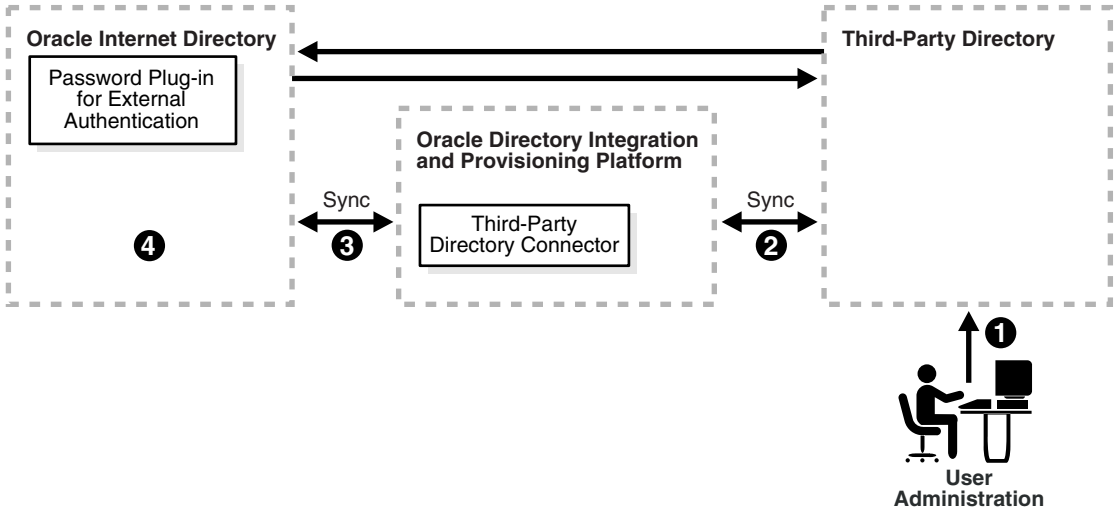
Third-Party Directory as the Central Directory

If a third-party directory is the central directory, then, once user, group, and realm objects are created, the third-party directory becomes the source of provisioning information for all Oracle components and other directories. In this case, Oracle Internet Directory is deployed to support Oracle components. To provide this

support, Oracle Internet Directory stores a footprint that enables it to identify entries in the third-party directory.

Figure 41–2 shows a typical deployment where a third-party directory is the central enterprise directory.

Figure 41–2 Interaction of Components with a Third-Party Directory as the Central Directory



Process for Provisioning of a User or Group

As Figure 41–2 shows, when a third-party directory is the central enterprise directory, typical provisioning of a user or group follows this process:

1. The user or group entry is created in the third-party directory.
2. At the next scheduled interval, the entry creation event is read by the third-party directory connector in the Oracle Directory Integration and Provisioning platform.
3. Following the mapping information in the integration profile, the user or group attributes in the third-party directory are mapped to the corresponding attributes in Oracle Internet Directory.
4. The user or group entry is created in Oracle Internet Directory.

Process for Modifying a User or Group Entry

An entry is modified in the third-party directory when:

- A new attribute gets added to the entry
- The value of an existing attribute is modified
- An existing attribute is deleted

When a third-party directory is the central enterprise directory, modification of a user or group entry follows this process:

1. The entry is modified in the third-party directory.
2. At the next scheduled interval, that entry modification event is read by the third-party directory connector in the Oracle Directory Integration and Provisioning platform,
3. Following the mapping information in the integration profile, the attribute in the third-party directory is appropriately mapped to the corresponding attribute in Oracle Internet Directory.
4. The user or group entry is modified in Oracle Internet Directory.

As Figure 41–2 shows, when a third-party directory is the central enterprise directory, modification of passwords happens asynchronously in the directory that serves as the password repository. This happens by using plug-ins.

Choose Where to Store Passwords

Regardless of which directory is the central enterprise directory, the password can be stored in one or both directories. There are advantages and disadvantages to each option. This section contains these topics:

- Advantages and Disadvantages of Storing the Password in One Directory
- Advantages and Disadvantages of Storing the Password in Both Directories

Advantages and Disadvantages of Storing the Password in One Directory

By reducing to one the number of points of possible attack, storing the password in only one directory can make the password more secure. Moreover, it eliminates synchronization issues when the password is modified.

On the other hand, storing the password in one directory provides a single point of failure for the entire network. If the directory that fails is a third-party one, then

even though user footprints are available in Oracle Internet Directory, users cannot access Oracle components.

Moreover, although storing passwords in only the central directory eliminates any possible synchronization issues, it requires you to enable applications to authenticate users to that directory. This involves using the appropriate plug-ins. For example, if you are using Microsoft Active Directory as both the central enterprise directory and the central password store, then you must enable applications to authenticate users to Microsoft Active Directory. You do this by using an external authentication plug-in. A similar mechanism is supported for authentication against SunONE Directory Server.

Note: Oracle components use password verifiers to authenticate users, and, when passwords are stored in the third-party directory, those verifiers are not stored in Oracle Internet Directory. On the other hand, if a password is modified by using an Oracle component, then the verifiers are both generated and stored in Oracle Internet Directory.

Advantages and Disadvantages of Storing the Password in Both Directories

If you decide to store the password in both directories, then passwords need to be synchronized, ideally in real-time.

In Oracle Internet Directory 10*g* (9.0.4), passwords are not synchronized in real time, but according to a schedule. This can mean an observable delay between the time the password is changed in the central directory and the time that the change is recorded in the other directory. In deployments with Oracle Internet Directory as the central directory, password values are synchronized on a regular basis from Oracle Internet Directory to the connected directory. This synchronization requires you to enable both the password policy of the realm and reversible encryption.

See Also:

- Chapter 15, "Password Policies in Oracle Internet Directory" for information about setting password policies
- Chapter 16, "Directory Storage of Password Verifiers" for information about reversible encryption

In general, password values are hashed. If both directories use the same hashing algorithm, then the hashed values can be synchronized as they are. For example,

suppose that you have an environment in which SunONE Directory Server and Oracle Internet Directory are integrated. Both of these directories support common hashing algorithms. Now, if the passwords are hashed and stored in SunONE Directory Server by using a hashing technique supported by Oracle Internet Directory, then synchronizing them from SunONE Directory Server to Oracle Internet Directory is the same as with any other attribute.

However if both directories do not support same hashing algorithm, then passwords must be synchronized in cleartext format only. For security reasons, password synchronization is possible with Oracle Internet Directory only in SSL Mode 2—that is, server-only authentication.

If Oracle Internet Directory is the source of truth, and if the hashing algorithm it supports is not supported by the other directory, then synchronization is still possible through SSL mode 2 (`sslmode=2`) when reversible password encryption is enabled.

If Microsoft Active Directory is the source of truth, then, when a password is modified in Microsoft Active Directory, a plug-in intercepts the password changes and stores the modified password in a new attribute, preferably in an encrypted form. That attribute can then be synchronized to Oracle Internet Directory. A similar process is required if Oracle Internet Directory is the central enterprise directory and central password store.

Note: In deployments where both directories do not use the same hashing algorithm, password synchronization is not available in an out-of-the-box installation of Oracle Internet Directory. You must configure it.

See Also: The following chapters for more detailed information about password synchronization:

- Chapter 42, "Integration with SunONE (iPlanet) Directory Server"
- Chapter 43, "Integration with the Microsoft Windows Environment"

The following chapter for information about plug-ins:

- Chapter 45, "Oracle Internet Directory Plug-in Framework"
- Chapter 47, "Setting Up the Customized External Authentication Plug-in"

Choose the Structure of the Directory Information Tree

At installation, each directory server creates a default domain and a default **directory information tree (DIT)** structure. When synchronizing with a third-party directory, you may want to create identical DIT structures on both directories. Otherwise, you need to do domain level mapping. If you choose domain-level mapping, then there are some limitations in synchronizing groups and other entries that have DNs in their attributes.

This section contains these topics:

- Create Identical DIT Structures on Both Directories
- Domain-Level Mapping and Limitations

Create Identical DIT Structures on Both Directories

Oracle Corporation recommends that you configure identical DITs on both directories. This enables all the user and group objects to be synchronized as they are, and spares you the cumbersome task of mapping entries with distinguished names in one directory to URLs in the other. It also spares you the performance problems that such mapping can cause.

To create identical DITs, first decide which directory is the central enterprise directory, and then change the DIT of the other one to match. Be sure to update the directory integration and provisioning profile to reflect the domain level rules.

To enable users to access Oracle applications through Oracle Application Server Single Sign-On, Oracle Corporation recommends that you identify the DIT as a separate identity management realm with its own authentication and authorization domain.

See Also: Chapter 19, "Deployment of Oracle Identity Management Realms" for information about identity management realms

Domain-Level Mapping and Limitations

If it is not feasible to have identical DITs on both directories, then you need to map the domains between Oracle Internet Directory and the connected directory. For example, suppose that all entries under the container `dc=mydir,dc=com` must be synchronized under `dc=myoid,dc=com` in Oracle Internet Directory. To achieve this, you specify it in the domain level mapping rules.

If the objective is to synchronize all users and groups, then all user entries can be synchronized with appropriate domain-level mapping. However, group entry synchronization may be both time consuming and carry some additional limitations. This section provides examples of both user and group synchronization when there is a domain-level mapping.

Example: User Entry Mapping

Suppose that, in a mapping file, the entries in the SunONE Directory Server have the format `uid=name,ou=people,o=iplanet.org`. Suppose further that the entries in Oracle Internet Directory have the format `cn=name,cn=users,dc=iplanet,dc=com`. Note that the naming attribute on SunONE Directory Server is `uid`, but on Oracle Internet Directory it is `cn`.

The mapping file has rules like these:

```
DomainRules
ou=people,o=iplanet.org: cn=users,dc=iplanet,dc=com: cn=%, cn=users,
dc=iplanet,dc=com
AttributeRules
Uid:1: :person:cn: :inetorgperson:
```

The value of `1` in the second column of the last line indicates that, for every change to be propagated from SunONE Directory Server to Oracle Internet Directory, the `uid` attribute must be present. This is because `uid` must always be available for constructing the DN of the entry in Oracle Internet Directory.

Example: Group Entry Mapping

When there is a domain-level mapping, synchronizing group entries is somewhat complex. The group memberships, which are DNs, must have valid DN values after synchronization. This means that whatever domain-level mapping was done for user DNs must be applied to group membership values.

For instance, suppose that the user DN values are mapped as follows:

```
ou=people,o=iplanet.org: cn=users,dc=iplanet,dc=com:
```

This implies that all the user entries under ou=people,o=iplanet.org are moved to cn=users,dc=iplanet,dc=com.

Group memberships need to be mapped as follows:

```
uniquemember: : : groupofuniquenames: uniquemember: :groupofuniquenames:
dnconvert(uniquemember)
```

For example, if the value of uniquemember is cn=testuser1,ou=people,o=iplanet.org, then it becomes cn=testuser1,cn=users,dc=iplanet,dc=com.

Moreover, if the value of uniquemember is cn=testuser1,dc=subdomain,ou=people,o=iplanet.org, then it becomes cn=testuser1,dc=subdomain,cn=users,dc=iplanet,dc=com.

This is a feasible solution as long as the naming attribute or RDN attribute remains the same on both the directories. However, if the naming attribute is different on different directories—as, for example, ou=people,o=iplanet.org:cn=users,dc=iplanet,dc=com:cn=%,cn=users,dc=iplanet,dc=com–then deriving the actual DNs for group memberships is not achievable through the given set of mapping rules. In this case, domain-level mapping for the uniquemember or other DN type attributes is not currently feasible.

If you want to synchronize group memberships, remember to keep the naming attribute in the source and destination directories the same.

See Also: "Format of the Mapping Rules Attribute" on page 33-7 for instructions on how to specify a mapping rule

Select the loginID Attribute

The loginID attribute contains the identity of the end user when logging into any Oracle component. This attribute is stored in Oracle Internet Directory as the value of the attribute orclcommonnicknameattribute, under the container cn=common,cn=products,cn=oracleContext,*identity_management_realm*.

By default, orclcommonnicknameattribute has uid as its value. This means that the identity used for login is stored in the uid attribute of the user entry.

If the connected directory has a specific attribute for login, then that attribute needs to be mapped to the right orclcommonnicknameattribute in Oracle Internet Directory. This needs to be one of the mapping rules in the mapping file for the connector associated with synchronizing with the third-party directory.

For example, suppose that you are synchronizing Oracle Internet Directory with Microsoft Active Directory, and that, in the latter, the login identifier is contained in the userPrincipalName attribute of the user entry. You would synchronize the value of the userPrincipalName attribute to Oracle Internet Directory, storing it in the uid attribute, which is the value of the orclcommonnicknameattribute

attribute. This mapping needs to be reflected in the mapping rules in the directory integration profile.

You can also use any other attribute for login. For example, if you want to use `employeeID` for logins, then mapping rules can be set accordingly. Doing this does not affect your configuration.

> **See Also:** "Configuring an Identity Management Realm by Using the Oracle Internet Directory Self-Service Console" on page 31-11 for instructions on setting the attribute for login name

Select the User Search Base

The user search context is represented by a multivalued attribute that lists all the containers under which users exist. Depending on your deployment, either set the user search context value to cover the entire user population, or add the container to the user search context attribute by using the Oracle Internet Directory Self-Service Console.

> **See Also:** "Configuring an Identity Management Realm by Using the Oracle Internet Directory Self-Service Console" on page 31-11 for instructions on setting the user search context

Select the Group Search Base

The group search context is represented by a multivalued attribute that lists all the containers under which groups exist. Depending on your deployment, either set the group search context value to cover all group entries, or add the container to the group search context attribute by using the Oracle Internet Directory Self-Service Console.

> **See Also:** "Configuring an Identity Management Realm by Using the Oracle Internet Directory Self-Service Console" on page 31-11 for instructions on setting the group search context

Decide How to Address Security Concerns

There are three main security concerns you need to consider:

- Access policies—The user and group search bases should be appropriately protected from the access of any malicious users.

- Synchronization—You can configure the Oracle directory integration and provisioning server to use SSL when connecting to Oracle Internet Directory and third-party directories. If you do this, then all information exchanged between the directory servers is secure.
- Password synchronization—Depending on the configuration, passwords can be synchronized. For instance, when Oracle Internet Directory is the central enterprise directory, password changes can be communicated to the connected directory.

  If passwords are to be synchronized, then Oracle Corporation recommends that you configure communication between the directories in SSL with server-only authentication. The sequence of steps to configure communication between connected directories in SSL is as follows:

  1. In the integration profile, to indicate that the mode of communication is SSL, configure the `connectedDirectoryURL` attribute in the form of `host:port:1`. Make sure the port number is the SSL port. The default SSL port number is 636.
  2. Generate a certificate from the connected directory. What is required is the trust point certificate from the server. You do not need to use any external certificate server to do this.
  3. Export the certificates to Base 64 encoded format.
  4. Import the certificates as trust points in the Oracle Wallet by using Oracle Wallet Manager.
  5. Specify the wallet location in the `odi.properties` file in `$ORACLE_HOME/ldap/odi/conf`.
  6. Store the wallet password by using the Directory Integration and Provisioning Assistant with the `wp` option.
  7. Start the Oracle directory integration and provisioning server in SSL mode.

Configuring Synchronization with a Third-Party Directory: Step by Step Guide

This section lists the steps in configuring a sample deployment scenario. Step "Step 4: Decide Whether to Create a New Identity Management Realm" through "Step 6: Select the Login Identifiers" involve configuring a new identity management realm and setting its parameters. might This can affect the behavior of Oracle Application Server Single Sign-On and any other middle-tier application already installed in the

environment. Consequently, make careful decisions at each step and verify the behavior of the applications.

> **See Also:** Chapter 19, "Deployment of Oracle Identity Management Realms" for more details on identity management realms and their role in Oracle Application Server.

This section contains these topics:

Step 1: Identify the Default Identity Management Realm in Oracle Internet Directory

To identify the default identity management realm in Oracle Internet Directory:

```
ldapsearch -p port -h host -D distinguished_name -w password
-b "cn=common, cn=products,cn=oraclecontext" -s base "objectclass=*"
```

```
defaultsubscriber
```

In this sample deployment, the default identity management realm in Oracle Internet Directory is `dc=us,dc=mycompany,dc=com`.

Step 2: Identify the User and Group Search Bases in Oracle Internet Directory

To identify the user and group search contexts in Oracle Internet Directory:

```
ldapsearch –p <port> -h <host> -D distinguished_name -w <passwd>
b "cn=common, cn=products,cn=oraclecontext, <Identity Management Realm>" -s base
"objectclass=*"
```

Note down the values for the `orclcommonusersearchbase` and `orclcommongroupsearchbase` attributes. These are the values which are shown in the Oracle Internet Directory Self-Service Console as User Search Context and Group Search Context.

In this sample deployment, the user and group search contexts in Oracle Internet Directory are:

```
orclcommonusersearchbase is : cn=users,  dc=us,dc=mycompany,dc=com
orclcommongroupsearchbase is : cn=groups,  dc=us,dc=mycompany,dc=com
```

Step 3: Identify the Naming Context on the Remote Directory

The default naming context is the root of the naming context under which the users are stored. Each directory has its own way of creating a default naming context.

If you are using Microsoft Active Directory, then you identify the default naming context by performing the following ldapsearch against that directory:

```
ldapsearch –p port -h host -D distinguished_name -w password -b "" –s base
"objectclass=*" defaultnamingcontext
```

Typically the DNs of users in Microsoft Active Directory are of the form `cn=user name, cn=users, defaultnamingcontext`.

Note that the users also can bind with names such as, `username@domain`.

For example, if the domain name is `newcompany.com`, then the default naming context is `dc=newcompany,dc=com`. The typical login identifier of a user is `user@newcompany.com`.

If you are using SunONE Directory Server, then you identify the naming contexts in SunONE Directory Server by performing the following ldapsearch against SunONE Directory Server:

```
ldapsearch -p port -h host -D distinguished_name -w password -b "" -s base
"objectclass=*" namingcontexts
```

Different sets of user entries reside in different subtrees. Choose the naming context that contains the objects to be synchronized.

Step 4: Decide Whether to Create a New Identity Management Realm

If the DITs on Oracle Internet Directory and the third-party directory are different, then it is better to create a new identify management realm. Do this by using either the Oracle Internet Directory Self-Service Console or the Oracle Internet Directory Configuration Assistant. On the other hand, if the third-party directory is Microsoft Active Directory in which the default naming context is `mycompany.com`, then you may not have to create the new identity management realm.

Step 5: Select the User Search Base and Group Search Base

How you do this depends on whether you created a mew identity management realm as discussed in the previous step.

If a new identity management realm has been created, then:

1. Select the user search base and the user creation context. Do this by using the Oracle Internet Directory Self-Service Console. Set the user search context to reflect the container under which users are stored in the third-party directory. This is described in "Configuring an Identity Management Realm by Using the Oracle Internet Directory Self-Service Console" on page 31-11.

   Follow the same approach to set the user creation context.

2. Select the group search base and the group creation context. Do this by using the Oracle Internet Directory Self-Service Console. Set the group search context to reflect the container under which groups are stored in the third-party directory. This is described in "Configuring an Identity Management Realm by Using the Oracle Internet Directory Self-Service Console" on page 31-11.

   Follow the same approach to set the group creation context.

If a new identity management realm has not been created, then, to enable user and group entries to be accessed by all Oracle components, you must modify the default parameters in the Oracle Internet Directory Self-Service Console. To do this:

1. In the User Search Context, enter the DN of the users container in the third-party directory, or enter the subtree of the containers specified in the search context. For example, enter either of the following:

   `cn=users,dc=myCompany,dc=com`

   `dc=myCompany,dc=com`.

2. In the Group Search Context, either enter the DN of the groups container in the third-party directory, or enter the subtree of the containers specified in the search context. For example, enter either of the following:

   `cn=groups,dc=myCompany,dc=com`

   `dc=myCompany,dc=com`

   See Also: "Configuring User Entries by Using the Oracle Internet Directory Self-Service Console" on page 31-14

Step 6: Select the Login Identifiers

The attribute used for login is `orclcommonnicknameattribute`. In the Oracle Internet Directory Self-Service Console, the field is named Attribute for Login Name. The default value is `UID`. Oracle Corporation recommends that you keep the default value. If this attribute is modified—for example, if it is changed to `mail`—then be sure that all entries under the container that you are working with have the `mail` attribute value populated. Otherwise, the user cannot login through Oracle Application Server Single Sign-On.

Step 7: Modify the Mapping File to Reflect the Changes You Have Made

The attributes you have just modified can require a change in the default mapping files. Look carefully at the various mapping rules and modify them according to the requirements. If the users and groups are under different containers, you may need to specify multiple set of domain rules in the same mapping file.

Default mapping rules for integration with SunONE Directory Server and Microsoft Active Directory are in the directory `$`*`ORACLE_HOME`*`/ldap/odi/conf`.

The important parameters to be modified are:

- Mapping rule for the `loginid` attribute
  - In the default profile for Microsoft Active Directory, the default mapping rule for the `loginid` attribute in the sample mapping file is:

    ```
    Userprincipalname: : :user: uid: : :inetorgperson
    ```

  - In the default profile for SunONE Directory Server, the `UID` is directly mapped to the `UID` attribute.

    This can be modified depending on which attribute is used for login. For example, to use `employeenumber` as the `loginid`, modify the mapping rule as follows:

    ```
    Employeenumber: : :user: uid: : :inetorgperson
    ```

- Mapping rule for the Kerberos login—To support Windows native authentication, Oracle Application Server Single Sign-On uses Kerberos login for the Windows environment. In such cases, a mapping rule is required for the Windows login. The attribute for the Kerberos login is `orclcommonkrbprincipalattribute` in the entry `cn=common,cn=public,cn=oraclecontext,`*`identity_management_realm`*. By default, it is set to `krbPrincipalName`.

  For integration with Microsoft Active Directory, the default mapping rule is:

  ```
  Userprincipalname: : :user: krbPrincipalName: : :orclUserV2.
  ```

  This rule maps the user principal name in Microsoft Active Directory to the Kerberos principal name. To support another value for Kerberos login, modify this rule.

  > **See Also:** *Oracle Application Server Single Sign-On Administrator's Guide* for information about support for Windows native authentication in Oracle Application Server Single Sign-On

Step 8: Create or Modify the Synchronization Profile with the New Set of Mapping Rules

To do this, use the Directory Integration and Provisioning Assistant.

```
dipassistant mp -profile profile_name odip.profile.mapfile=relative_path_name_
of_mapping_file
```

Step 9: Configure Access Control

Configure access control to various containers in either of the following:

- The profile `orclodipagentname=`*`profile_name`*`,cn=subscriber profile,cn=changelog subscriber,cn=oracle internet directory'`

- The group `cn=odipgroup,cn=odi,cn=oracle internet directory`

A sample ACI is available in *ORACLE_HOME*`/ldap/odi/samples/commonaci.ldif`. This sample contains the following attributes, all of which have the same values:

- `UserSearchBase`
- `GroupSearchBase`
- `UserCreateBase`
- `GroupCreateBase`

You can use Oracle Directory Manager to set ACIs to these containers.

Step 10: Bootstrap the Directory by Using the Directory Integration and Provisioning Assistant

To bootstrap the directory, use the `bootstrap` command in the Directory Integration and Provisioning Assistant.

See Also:

- Chapter 37, "Bootstrapping of a Directory in the Oracle Directory Integration and Provisioning Platform"
- "The Directory Integration and Provisioning Assistant" on page A-107 for instructions on using the `bootstrap` command of the Directory Integration and Provisioning Assistant

Step 11: Update the Last Change Number for Synchronization

To do this, enter:

```
dipassistant mp -profile profile_name -updlcn
```

The Directory Integration and Provisioning Assistant determines the connected directory by reading the directory integration profile.

Step 12: Enable the Profile by Using Either Oracle Directory Manager or the Directory Integration and Provisioning Assistant

You can do this by using either Oracle Directory Manager of the Directory Integration and Provisioning Assistant.

See Also:

- "Registering a Profile by Using Oracle Directory Manager" on page 33-20 for instructions on doing this by using Oracle Directory Manager
- "Registering and Deregistering a Synchronization Profile by Using the Directory Integration and Provisioning Assistant" on page 33-22 for instructions on doing this by using the Directory Integration and Provisioning Assistant

Step 13 (Optional): Enable the External Authentication Plug-in for Password Synchronization

If you need to synchronize password changes from Oracle Internet Directory to the third-party directory, then enable the external authentication plug-in by doing the following:

- Enable the password policy in the identity management realm. You can do this by using either the Oracle Internet Directory Self-Service Console or Oracle Directory Manager.
- Enable reversible password encryption by setting the `orclpwdencryptionenable` attribute to `TRUE`.

When passwords are synchronized to directories that do not support the hashing technique used by Oracle Internet Directory, synchronization can be done only by using the SSL mode 2 (`sslmode=2`).

See Also:

- "Managing Password Policies by Using the Self-Service Console" on page 15-10
- "Managing Password Policies by Using Oracle Directory Manager" on page 15-6
- "Storing and Managing Password Verifiers for Authenticating to Oracle Internet Directory" on page 16-2 for information about enabling reversible encryption

Step 14: Start the Oracle Directory Integration and Provisioning Server

Do this by following the instructions in "Starting the Oracle Directory Integration and Provisioning Server" on page A-11.

Note: To synchronize passwords, start the Oracle Directory Integration and Provisioning platform with `sslmode=2`—that is, server-only authentication.

Limitations of Third-Party Integration in Oracle Internet Directory 10*g* (9.0.4)

Oracle Internet Directory 10*g* (9.0.4) does not support the synchronization of the schema and ACLs. If you are changing the schema or ACLs, then you must apply the changes manually. The schemasync tool is available for this purpose.

See Also: "The schemasync Tool Syntax" on page A-125 for information about the SchemaSync tool

42

Integration with SunONE (iPlanet) Directory Server

This chapter explains how to integrate the Oracle Application Server infrastructure with SunONE Directory Server (Netscape Directory Server and iPlanet Directory Server) by using the SunONE connector in the Oracle Directory Integration and Provisioning platform.

> **Note:** This chapter assumes that you have read Chapter 41, "Considerations for Integrating with Third-Party Directories" and made the necessary deployment decisions and basic configurations.

This chapter contains these topics:

About the SunONE Connector

The SunONE connector includes a synchronization component that is driven by the Oracle directory integration and provisioning server. This component maintains consistency between the directories by:

- Importing data and incremental changes from an SunONE Directory Server into
- Exporting data and incremental changes from Oracle Internet Directory into an SunONE Directory Server

The SunONE Directory Server and Oracle Internet Directory support similar hashing techniques for storing passwords. If both the directories are configured to use the same hashing algorithm, and the mapping rules are configured appropriately, then the SunONE connector can synchronize passwords as it does any other attribute. To store the password in the SunONE Directory Server, use the SunONE Directory Server external authentication plug-in discussed in this chapter.

Note: Oracle Internet Directory 10*g* (9.0.4) can synchronize with Netscape Directory Server Releases 4.13 and SunONE (iPlanet) Directory Server Releases 5.0 and 5.1.

See Also:

- "Hashing Schemes for Creating Password Verifiers" on page 16-3 for a list of hashing algorithms supported by Oracle Internet Directory
- "Task 4: (Optional) Configure the SunONE Directory Server External Authentication Plug-in" on page 42-11 for instructions on configuring the SunONE (iPlanet) Directory Server external authentication plug-in

SunONE Directory Server Integration Concepts

This section contains these topics:

- Synchronization Between Oracle Internet Directory and SunONE Directory Server
- The SunONE Directory Server External Authentication Plug-in

Synchronization Between Oracle Internet Directory and SunONE Directory Server

Synchronization with SunONE Directory Server is based on reading incremental changes from the source directory to the destination directory. If changes are to be made in both directories, then both directories need to have change logging enabled.

See Also:

- Starting an Oracle Directory Server Instance on page A-7 for instructions on how to start an Oracle directory server with change logging enabled
- SunONE Directory Server documentation for instructions on how to start the SunONE Directory Server with change logging enabled

If you want to synchronize passwords, then be sure that the hashing technique used by SunONE Directory Server is also supported by Oracle Internet Directory. The current hashing technique enabled in Oracle Internet Directory, can be obtained by doing a base search in Oracle Internet Directory as follows:

```
ldapsearch  -h host -p port_number -b '' -s base 'objectclass=*'
orclcryptoscheme
```

The SunONE Directory Server External Authentication Plug-in

Oracle components are clients of Oracle Internet Directory. However, in an integrated environment, you have the option of storing security credentials for those components in an external repository —in this case, SunONE Directory Server—rather than in Oracle Internet Directory. When security credentials are stored in an external repository, user authentication to an Oracle component happens in the external repository and not in Oracle Internet Directory.

To communicate with the external repository, the Oracle component relies on the Oracle directory server. The Oracle directory server, in turn, uses a plug-in that can access the external repository. The entire authentication process is transparent to the Oracle components, which perceive all the LDAP requests as being handled by the Oracle directory server.

Types of External Authentication

To verify a user's security credentials, an Oracle component can, by way of the Oracle directory server, send to the external repository a simple bind with a request for one of the following:

- Non-SSL ldapbind
- SSL ldapbind
- ldapcompare

How Authentication to an External Repository Works

When an Oracle directory server has the plug-in configured and enabled, the following process occurs to authenticate a user to an Oracle component.

1. The user seeks access to an Oracle component.
2. The Oracle component, which is a client of Oracle Internet Directory, receives the authentication request, and passes to the Oracle directory server either an ldapbind or ldapcompare request.
3. The Oracle directory server passes the control to the plug-in.
4. The plug-in issues the request to the external repository.
5. The plug-in obtains the results of that request and passes the results back to the Oracle directory server.
6. The Oracle directory server passes the results back to client application, which then grants or denies access to the user.

Configuring the SunONE Connector

This section explains the tasks to configure the SunONE connector. It contains these topics:

- Task 1: Configure the Integration Profile for the SunONE Connector
- Task 2: Configure Access Control Lists
- Task 3: Prepare Both Directories for Synchronization
- Task 4: (Optional) Configure the SunONE Directory Server External Authentication Plug-in
- Task 5: Start the Synchronization

Task 1: Configure the Integration Profile for the SunONE Connector

Integration profile templates for synchronization with the SunONE Directory Server are created in the Oracle directory server as a part of the installation process.

There are two default integration profiles:

- `iPlanetImport`—for importing entries and changes from the SunONE Directory Server by using the directory synchronization approach
- `iPlanetExport`—for exporting changes from Oracle Internet Directory to SunONE Directory Server

These are simply templates to be customized to meet the needs of your deployment.

Customizing the Default Integration Profiles

To customize the default integration profiles, you can use a shell script, the Directory Integration and Provisioning Assistant, or Oracle Directory Manager. Configure separate profiles for import and export operations.

Configuring the Default Integration Profile through the script iplconfig.sh Use this method when:

- The SunONE Directory Server has no custom schema changes to the objects to be synchronized—that is, the user and group object attributes and object classes are the default ones
- No custom schema elements have been added to the user or group object attributes and object classes

At the end of synchronization, user and group objects synchronized from the SunONE Directory Server are visible to Oracle components integrated with the Oracle Application Server infrastructure.

The script `iplconfig.sh` resides in `$`*`ORACLE_HOME`*`/ldap/odi/admin`. It prompts you for the following:

- Oracle Internet Directory super user DN and password
- SunONE Directory Server URL (*`host`*`:`*`port`*)
- SunONE Directory Server user account and password to be used by the SunONE connector
- SunONE Directory Server domain to be synchronized

Once you have entered the parameter values, `iplconfig.sh` invokes the Directory Integration and Provisioning Assistant to set up the SunONE Directory Server connection information and mapping rules information in the default SunONE Directory Server integration profiles.

Configuring the Default Integration Profile by Using the Directory Integration and Provisioning Assistant or Oracle Directory Manager Use this method when:

- The SunONE Directory Server default has been customized according to the deployment requirements—that is, the user or group objects have custom schema elements
- Oracle Internet Directory objects have custom schema elements
- The objects and attributes to be synchronized between the two directories are not the default ones

To configure the directory integration profile by using this method, follow these steps:

1. Configure the mapping rules as described in "Configuring Mapping Rules" on page 42-7.
2. Update the default parameters as described in "Updating the Default Parameters" on page 42-7.
3. Bootstrap the directories as described in "Task 3: Prepare Both Directories for Synchronization" on page 42-10. The default mapping file for bootstrapping should be like the `iplanetimp.map.master` file. When you make changes, use this file as the sample.
4. Configure password synchronization. The default mapping rules are not appropriate for password synchronization between the SunONE Directory Server and Oracle Internet Directory.

   If Oracle Internet Directory and the SunONE Directory Server use the same password hashing technique, then insert the following mapping rule to the mapping file and upload the mapping file to the profile.

   ```
   Userpassword: : :person:userpassword: :person
   ```

   If the two directories do not use the same hashing technique, then the same mapping rule works when the Oracle directory integration and provisioning server and the directory integration profile are configured in SSL mode 2—that is, server-only authentication.

Configuring Mapping Rules

The default profiles have the default mapping rules for mapping the user and group attributes and object classes in SunONE Directory Server to those on Oracle Internet Directory. These mapping rules assume that no user- and group-specific schema changes have been made to either directory after installation. If there are such changes, then they must be appropriately reflected in the mapping files.

To verify and modify the mapping rules, do the following:

1. Decide which domains, or containers, you want to synchronize. In the case of SunONE Directory Server, the container to be specified for synchronization can be any naming context in the directory.
2. Decide on the objects—that is, the types of entries—to be synchronized. In an identity management environment these are typically user and group entries.
3. Identify the attributes and how you want to map them between the directories during synchronization.
4. Generate a mapping file with appropriate mapping rules.

See Also: "Format of the Mapping Rules Attribute" on page 33-7 for instructions on creating mapping rules and for sample mapping files

Updating the Default Parameters

Once the mapping file is generated, you can update the parameters in the default integration profile by using either Oracle Directory Manager or the Directory Integration and Provisioning Assistant. Table B–20 on page B-18 lists the attributes in the default integration profile for third-party directories. Some of those attributes,

listed in Table 42–1, have values specific to integration with the SunONE Directory Server.

Table 42–1 Default Attribute Values in the SunONE Directory Server Integration Profile

Attribute	Value
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.
Connected Directory URL (`orclOdipConDirURL`)	Connect details required to connect to the connected directory. This parameter refers to the host name and port number as *`host`*`:`*`port`*`:`*`sslmode`*. To connect by using SSL, enter *`host`*`:`*`port`*`:1`. Make sure the certificate to connect to the directory is stored in the wallet, the location of which is specified in the file `odi.properties`. **Note:** To connect to SunONE Directory Server by using SSL, the server certificate needs to be loaded into the wallet. **See Also:** The chapter on Oracle Wallet Manager in *Oracle Advanced Security Administrator's Guide*
Mapping Rules (`orclOdipAttributeMappingRules`)	Attribute for storing the mapping rules. Store the mapping rules in a file by using the Directory Integration and Provisioning Assistant or the `ldapuploadagentfile.sh` tool. **See Also:** ■ "Mapping Rules and Formats" on page 33-5 ■ "Format of the Mapping Rules Attribute" on page 33-7 ■ "The Directory Integration and Provisioning Assistant" on page A-107

Table 42–1 (Cont.) Default Attribute Values in the SunONE Directory Server Integration Profile

Attribute	Value
Connected Directory Account (`orclodipConDirAccessPassword`)	Password to be used by the user specified in the `orclOdipConDirAccessAccount` attribute to connect to the connected directory. For the SunONE synchronization connector, it is the valid bind password in the SunONE Directory Server.
Connected Directory Account (`orclodipConDirAccessAccount`)	If the changes are to be imported from SunONE Directory Server to Oracle Internet Directory, then this user account should have privileges to read the SunONE Directory Server change log container. If the changes in Oracle Internet Directory are to be exported to SunONE Directory Server, then the user must have privileges to add and modify in the synchronization domain. **Note:** Create a user account in SunONE Directory Server exclusively for the SunONE connector for synchronizing.
Agent Execution Command (`orclodipAgentExeCommand`)	This field must be empty.

See Also:

- "Registration of Connectors into the Oracle Directory Integration and Provisioning Platform" on page 33-7 for the required steps and a general description of each attribute you must set
- "The Directory Integration and Provisioning Assistant" on page A-107

Task 2: Configure Access Control Lists

Set up appropriate ACLs allowing read, add, or modify access rights on the subscribed domains.

During import operations, you would privilege the Oracle Internet Directory user `orclodipagentname=iPlanetImport,cn=subscriber profile,cn=changelog subscriber,cn=oracle internet directory` to update the subscribed domain in Oracle Internet Directory.

For example, assuming that no ACLs are applied to the domain of interest, the following LDIF sample can be used. In this file, the domain of interest is *`Synchronization_domain_in_OID`*.

```
ACL in OID:
```

```
dn: Synchronization_domain_in_OID
changetype: modify
add: 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)"
```

On the other hand, the privileges can also be granted to the group `cn=odipgroup,cn=odi,cn=oracle internet directory` of which the profile is a member. However, remember that, when privileges are granted to the group, all members of the group are, intentionally or not, granted privileges.

During import operations, the user specified by the Connected Directory Account attribute in the integration profile must have:

- Write access to the target container in Oracle Internet Directory
- Read access to the change log and source container in the SunONE Directory Server

During export operations, the user specified by the Connected Directory Account attribute in the integration profile must have:

- Write access to the target container in the SunONE Directory Server
- Read access to the change log and source container in the SunONE Directory Server

> **See Also:** SunONE Directory Server documentation for instructions on how to apply ACLs on the SunONE Directory Server change log container and the SunONE Directory Server subscribed domain

Task 3: Prepare Both Directories for Synchronization

Follow these steps:

1. Before the start of the synchronization, make the data in the domains of interest to be equivalent. This can be achieved by the Directory Integration and Provisioning Assistant with the bootstrap option. Bootstrapping is described in Chapter 37, "Bootstrapping of a Directory in the Oracle Directory Integration and Provisioning Platform".

2. If you have used LDIF file-based bootstrapping, then you must initialize the `lastchangenumber` value. You can do this by using the Directory Integration and Provisioning Assistant:

   ```
   dipassistant mp -profile profile_name -updlcn
   ```

3. At the end of bootstrapping, 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 change log enabled by using the **OID Control Utility**.

   Similarly, verify that change logging is enabled in SunONE Directory Server.

   See Also:

   - "The Directory Integration and Provisioning Assistant" on page A-107
   - "Starting and Stopping an Oracle Directory Server Instance" on page A-7 for a description of the OID Control Utility

Task 4: (Optional) Configure the SunONE Directory Server External Authentication Plug-in

If you are storing passwords in SunONE Directory Server, then you must use the SunONE Directory Server external authentication plug-in to authenticate SunONE Directory Server users from Oracle Internet Directory.

This section tells how to install, delete, enable, and disable the SunONE Directory Server external authentication plug-in by using the command line. You can perform these operations, except for installation, by using Oracle Directory Manager as described in "Registering and Managing Plug-ins by Using Oracle Directory Manager" on page 45-5.

Note: The SunONE Directory Server external authentication plug-in can be configured to authenticate to only one single SunONE Directory Server.

Installing the SunONE Directory Server External Authentication Plug-in

To install the plug-in:

1. Execute `$ORACLE_HOME/ldap/admin/oidspipi.sh`.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

To execute `oidspipi.sh`, enter:

```
cd $ORACLE_HOME/ldap/admin
oidspipi.sh
```

If you are using the Windows operating system, then execute oidspipi.sh after you have installed the UNIX emulation utility by entering:

```
sh oidspipi.sh
```

2. Enter the SunONE Directory Server host name. This is the SunONE Directory Server to which you are going to synchronize. This value is required.
3. Choose whether to use an SSL connection.

   When specifying the wallet location on the Microsoft Windows operating system, add an additional backslashes (\). For example, if the wallet location is `D:storage\wallet`, then enter `D:\\storage\\wallet`.
4. Enter the SunONE Directory Server port number.
5. Enter the database connect string.
6. Enter the ODS password. The default ODS password is the same as that set for the Oracle Application Server administrator during installation.
7. Enter Oracle directory server host name. This value is required.
8. Enter Oracle directory server port number. The default port is 389.
9. Enter the password of the Oracle administrator (`orcladmin`). This value is required.
10. Enter the distinguished name of the container to which the plug-in needs to be applied. Every entry in this container will be authenticated against SunONE Directory Server. Note that this need not necessarily be the User Search Base supplied in Oracle Internet Directory Self-Service Console. All the users under

this search base are authenticated externally to the SunONE Directory Server. If more than one value is specified, then use semi-colons (;) to separate them.

11. Enter the Plug-in Request Group DN. For security reasons, the plug-in can be invoked only by users belonging to this group. For example, suppose that the Oracle Application Server Single Sign-On administrators are in the group `cn=OracleUserSecurityAdmins,cn=Groups,cn=OracleContext`. If you enter this value for the Plug-in Request Group DN, then only requests coming from Oracle Application Server Single Sign-On administrators can trigger the external authentication plug-in. You can enter multiple DN values. Use a semicolon (;) to separate them. This value is not required, but, for security purposes, it should be specified.

12. Enter the value of the entry that is to be excluded from authentication to SunONE Directory Server. This value is the exception to item 10 on page 42-12. You need to enter the value in the standard ldapsearch filter format. For example, if you specify the value `(&(objectclass=inetorgperson)(cn=orcladmin))`, then any entry under the user container specified in item 10 that has the `cn=orcladmin` and `objectclass=inetorgperson` attribute value will not be authenticated to SunONE Directory Server.

13. Specify whether you want to back up the SunONE Directory Server for failover.

Deleting the SunONE Directory External Authentication Plug-in

To delete the SunONE Directory Server plug-in by using Oracle Directory Manager, follow the instructions in "Deleting a Plug-in by Using Oracle Directory Manager" on page 45-6.

To delete the SunONE Directory Server plug-in by using command-line tools, use these commands:

```
ldapdelete -h host -p port -D cn=orcladmin -w password
"cn=ipwhencompare,cn=plugin,cn=subconfigsubentry"

ldapdelete -h host -p port -D cn=orcladmin -w password
"cn=ipwhenbind,cn=plugin,cn=subconfigsubentry"
```

Enabling the SunONE Directory External Authentication Plug-in

To enable the SunONE Directory external authentication plug-in by using Oracle Directory Manager, follow the instructions in "Editing a Plug-in by Using Oracle Directory Manager" on page 45-5 and set the Plug-in Enable field to `1`.

To enable the SunONE Directory Server external authentication plug-in by using command-line tools, enter the following commands:

```
ldapmodify -h host_name -p port_number -D cn=orcladmin -w password <<EOF
dn: cn=ipwhencompare,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 1
EOF


ldapmodify -h host_name -p port_number -D cn=orcladmin -w password <<EOF
dn: cn=ipwhenbind,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 1
EOF
```

Disabling the SunONE Directory Server External Authentication Plug-in

To disable the SunONE Directory Server external authentication plug-in by using Oracle Directory Manager, follow the instructions in "Editing a Plug-in by Using Oracle Directory Manager" on page 45-5 and set the Plug-in Enable field to `0`.

To disable the SunONE Directory Server external authentication plug-in by using command-line tools, enter the following commands:

```
ldapmodify -h host_name -p port_number -D cn=orcladmin -w password <<EOF
dn: cn=ipwhencompare,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 0
EOF


ldapmodify -h <host> -p <port> -D cn=orcladmin -w <password> <<EOF
dn: cn=ipwhenbind,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 0
EOF
```

Enabling and Disabling SunONE Directory External Authentication Plug-in Debugging

If you are experiencing unknown errors, the you can enable the plug-in debugging. To do this, enter:

```
sqlplus ods/odspassword @$ORACLE_HOME/ldap/admin/oidspdon.pls
```

To check the plug-in debugging log, enter:

```
sqlplus ods/ods
select * from plg_debug_log order by id;
```

To delete the plug-in debugging log, enter:

```
sqlplus ods/ods
truncate table plg_debug_log
```

To disable the plug-in debugging, enter:

```
sqlplus ods/ods @$ORACLE_HOME/ldap/admin/oidspdof.pls
```

Note: If you need to change the plug-in setup—that is, the information you entered in the installation steps—then you can rerun the installation script. Before you rerun the script, delete the SunONE Directory external authentication plug-in by following the instructions in "Deleting the SunONE Directory External Authentication Plug-in" on page 42-13.

See Also:

- "Protection of User Passwords for Directory Authentication" on page 12-8 for a list of the hashing algorithms that Oracle Internet Directory supports for password protection
- SunONE Directory Server documentation for instructions on how to set the appropriate hashing algorithm for passwords in SunONE Directory Server

Task 5: Start the Synchronization

To start synchronization:

1. Enable the profile by setting the `profileStatus` attribute to `ENABLE` in either Oracle Directory Manager or the Directory Integration and Provisioning Assistant
2. Start the Oracle directory integration and provisioning server by using the OID Control Utility (oidctl) with the appropriate configuration set entry in which the profile is stored.

The Synchronization Process

The synchronization process is as follows:

1. In an import operation, the SunONE connector extracts all the changes from the SunONE Directory Server based on the value specified in the `orclodipConDirLastAppliedChgNum` attribute. It then applies them to Oracle Internet Directory.

   In an export operation, the SunONE connector extracts all the changes from Oracle Internet Directory based on the `orclodipLastAppliedChangeNumber` and applies them to the SunONE Directory Server.
2. Once all the changes are read and applied, the appropriate attribute—either `orclodipConDirLastAppliedChgNum` or `orclodipLastAppliedChangeNumber`—is updated.
3. After the execution is finished, the directory integration and provisioning server updates the execution status attributes.

Troubleshooting Synchronization with the SunONE Directory Server

This section contains these topics:

Location of Error Message File

The Oracle directory integration and provisioning server stores error messages in the appropriate file, as described in Table 35–3 on page 35-11.

How to Debug the SunONE Connector

You can debug the SunONE connector by using the oditest utility.

- To troubleshoot the SunONE import connector, run oditest with `AgentName` as `IplanetImport` and look at the `iPlanetImport.trc` and `iPlanetImport.aud` files.
- To troubleshoot the default SunONE export connector, run the oditest utility with `AgentName` as `IplanetExport` and look at the `iPlanetExport.trc` and `iPlanetExport.aud` files.

> **See Also:** "Troubleshooting Synchronization in the Oracle Directory Integration and Provisioning Platform" on page 33-23 for instructions on using the oditest utility

Supported Configurations for Integrating with SunONE Directory Server

In a deployment with Oracle Internet Directory as the central directory, the following configurations are supported:

- Identical DITs on both directories
- Synchronization by using domain mapping
- Password synchronization. In this environment, synchronization ensures only the creation of footprints on the SunONE Directory Server. Any other configuration changes required to access the user or group entries must be specifically handled by the deployment.

In a deployment with SunONE Directory Server as the central repository, the following configurations are supported:

- Identical DITs on both directories
- Synchronization by using domain mapping
- Password synchronization
- Plug-in-based authentication from Oracle Internet Directory

43

Integration with the Microsoft Windows Environment

This chapter explains how to configure synchronization with Microsoft Active Directory and to integrate the Oracle Application Server infrastructure with Microsoft Windows NT 4.0. You do this by using the Active Directory connector in the Oracle Directory Integration and Provisioning platform.

This chapter contains these topics:

Note: This chapter assumes that you have read Chapter 41, "Considerations for Integrating with Third-Party Directories" and made the necessary deployment decisions and basic configurations. After making the basic configurations, be sure that the complete infrastructure is working properly before starting synchronization.

See Also: "Oracle Internet Directory Frequently Asked Questions" on the Oracle Technology Network at `http://www.otn.oracle.com`.

About the Active Directory Connector

The Active Directory Connector comprises:

- A synchronization component, which is driven by the Oracle directory integration and provisioning server. This component maintains consistency between directory entries by importing data and incremental changes between a Microsoft Active Directory Server and Oracle Internet Directory. Note that, individual profiles need to be configured explicitly for import and export.
- A plug-in component to support authentication. This component is required if the deployment stores the password in Microsoft Active Directory.

Microsoft Active Directory and Oracle Internet Directory do not support similar hashing techniques for storing the passwords. Oracle Internet Directory cannot read passwords from Microsoft Active Directory. To store the password in Active directory server, use the Active Directory external authentication plug-in discussed in this chapter.

Microsoft Active Directory Integration Concepts

This section contains these topics:

Synchronization Between Oracle Internet Directory and Microsoft Active Directory

You can integrate Oracle Internet Directory with various configurations of Microsoft Active Directory. This section discusses some of the more typical Microsoft Active Directory configurations with which you can integrate. Specifically, it discusses:

- A single domain in Microsoft Active Directory
- Multiple domains in Microsoft Active Directory
- A forest in Microsoft Active Directory

In configuring the Oracle Directory Integration and Provisioning platform to synchronize in any of these configurations, it is necessary to understand the mode of Microsoft Active Directory synchronization.

Microsoft Active Directory provides various ways of tracking changes made to its directory contents. The directory synchronization connector, which is a component

of the Oracle Directory Integration and Provisioning platform, can synchronize by using either:

- The DirSync control-based approach for complete directory synchronization
- The USNChange-based approach for subtree synchronization

In each of these approaches, the directory from which changes are to be derived is polled at scheduled intervals.

Each approach has advantages and disadvantages depending on your deployment. Table 43–1 compares and contrasts the two approaches to synchronization.

Table 43–1 Comparing and Contrasting the DirSync Approach with the USNChanged Approach

Consideration	DirSync Approach	USNChanged Approach
Change key	Presents changes to the `ObjectGUID`—the unique identifier of the entry	Presents changes to the distinguished name. The `ObjectGUID` is used to keep track of modifications of the RDN.
Changes to multivalued attributes	Reflects incremental changes made to multivalued attributes as a complete replacement of the attribute value. This might cause unnecessary traffic on the network.	Reflects incremental changes made to multivalued attributes as a complete replacement of the attribute value. This might cause lot of unnecessary traffic on the network.
Error handling	If synchronization aborts, starts the next cycle from the current place. This requires keeping count of the number of changes applied during any synchronization operation. Otherwise, some changes are applied again.	Does not require synchronization to be atomic. If synchronization of a particular entry fails, then the next synchronization cycle can start from the current or next entry.
Information in the search results	Provides search results consisting of only the changed attributes and the new values. Application of these changes to Oracle Internet Directory is very easy.	Provides search results consisting of the complete changed entry. All the attribute values are compared to the old values stored in Oracle Internet Directory and applied, only if it has changed. This can be time consuming.
Monitoring of applied changes	When queried for changes in the directory, presents incremental changes based on a cookie value that identifies the state of the directory. Because the cookie is a binary value, changes over a period of time cannot be selectively ignored.	The changes are queried in the directory based on the USNChanged attribute, which is an Integer. It is very easy to modify the value if required.

Table 43–1 (Cont.) Comparing and Contrasting the DirSync Approach with the USNChanged Approach

Consideration	DirSync Approach	USNChanged Approach
Privileges required for the synchronizing user	Requires the user to have the `SE_SYNC_AGENT_NAME` privilege, which enables reading all objects and attributes in Microsoft Active Directory regardless of the access protections on the objects and attributes	No special privileges required. The user must have privileges to read and write in the specific container.
Support of multiple domains	Requires separately connecting to the different domain controllers to read changes made to the entries in different domains	Enables the user to read changes made to the multiple domains by connecting to the Global Catalog Server. In this case, deletions cannot be synchronized.
Synchronization from a replicated directory in case of failover	Can be continued as it is. The synchronization key is the same when connecting to a replicated environment.	Requires the change number to be updated before starting synchronization with the failover directory
Synchronization scope	Reads all the changes made in the directory, filters out the changes made to the required entries, and propagates to Oracle Internet Directory	Makes it possible to look for changes in any specific subtree
Two-way Synchronization	For two-way synchronization, requires configuring an import profile and an export profile for each of the domain controllers	For two way synchronization, requires one profile for importing changes from all the domain controllers, and individual profiles to export changes to each of the domain controllers
Usability in an environment with multiple Microsoft Active Directory servers behind a load balancer	Connect to a specific Microsoft Active Directory node, preferably a Global Catalog Server	Connect to a specific Microsoft Active Directory node

The Active Directory External Authentication Plug-in

Oracle components are clients of Oracle Internet Directory. However, in an integrated environment, security credentials for those components can be stored not in Oracle Internet Directory but in an external repository—for example, either Microsoft Active Directory or Microsoft Windows NT. When security credentials are stored in an external repository, user authentication to an Oracle component happens not in Oracle Internet Directory but in the external repository.

To communicate with the external repository, the Oracle component relies on the Oracle directory server. The Oracle directory server, in turn, uses a plug-in that can access the external repository. The entire authentication process is transparent to the Oracle components, which perceive all the LDAP requests as being handled by the Oracle directory server.

Types of External Authentication

To verify a user's security credentials, an Oracle component can, by way of the Oracle directory server, send to the external repository a simple bind with a request for one of the following:

- Non-SSL ldapbind
- SSL ldapbind
- ldapcompare

How Authentication to an External Repository Works

When an Oracle directory server has the plug-in configured and enabled, the following process occurs to authenticate a user to an Oracle component.

1. The user seeks access to an Oracle component.
2. The Oracle component, which is a client of Oracle Internet Directory, receives the authentication request, and passes to the Oracle directory server either an ldapbind or ldapcompare request.
3. The Oracle directory server passes the control to the plug-in.
4. The plug-in issues the request to the external repository.
5. The plug-in obtains the results of that request and passes the results back to the Oracle directory server.
6. The Oracle directory server passes the results back to client application, which then grants or denies access to the user.

Integration of Oracle Internet Directory and Microsoft Active Directory with a Single Domain

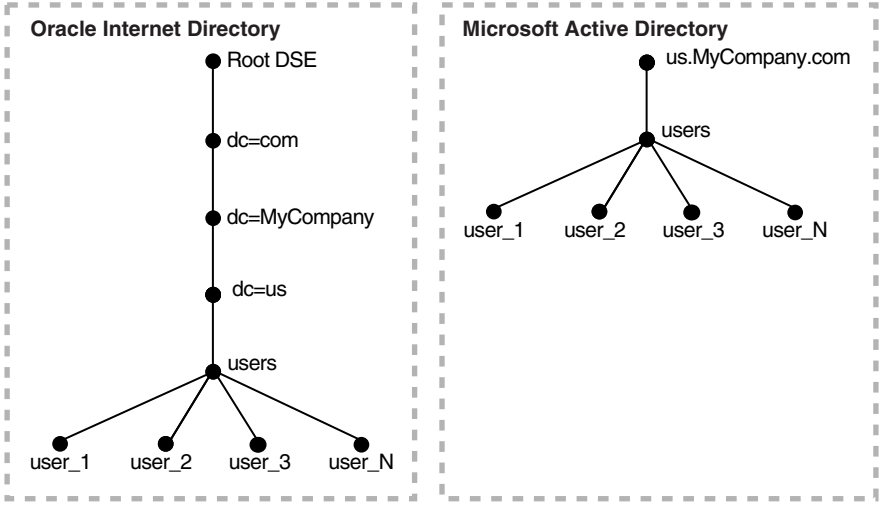
This section provides an example of how a Microsoft Active Directory with a single domain is mapped to Oracle Internet Directory, and explains the necessary configurations. It contains these topics:

- Example: Integration of Oracle Internet Directory and Microsoft Active Directory with a Single Domain
- Configuring Integration of Oracle Internet Directory with a Microsoft Active Directory with a Single Domain

Example: Integration of Oracle Internet Directory and Microsoft Active Directory with a Single Domain

Figure 43–1 shows an example of how a single domain in Microsoft Active Directory is mapped to a DIT in Oracle Internet Directory.

Figure 43–1 Integration of Oracle Internet Directory with a Single Domain in Microsoft Active Directory



In Figure 43–1, the node `us.MyCompany.com` in Microsoft Active Directory maps to `dc=us,dc=MyCompany,dc=com` in Oracle Internet Directory. The object

container—in this case, the `users` container in Microsoft Active Directory—is reflected in Oracle Internet Directory.

Configuring Integration of Oracle Internet Directory with a Microsoft Active Directory with a Single Domain

Whatever your deployment decisions, configuring integration with Microsoft Active involves:

Task 1: Configure the Integration Profile

Task 2: Configure Access Control Lists

Task 3: Prepare Both Directories for Synchronization

Task 4: (Optional) Configure the Active Directory External Authentication Plug-in

Task 5: Start the Synchronization

Task 1: Configure the Integration Profile

To configure the integration profile, you customize parameters in default integration profiles.

About the Default Integration Profiles

During installation, default integration profiles for synchronizing with the Microsoft Active Directory Server are created in Oracle Internet Directory. There are three integration profiles:

- `ActiveImport`—The profile for importing entries and changes from Microsoft Active Directory by using the DirSync approach
- `ActiveChgImp`—The profile for importing entries and changes from Microsoft Active Directory by using the change number-based approach
- `ActiveExport`—The profile for exporting changes from Oracle Internet Directory to Microsoft Active Directory

These are simply templates to be customized to meet the needs of your deployment.

Customizing the Default Integration Profiles

To customize the default integration profiles, you can use the shell script `adprofileconfig.sh`, the Directory Integration and Provisioning Assistant, or

Oracle Directory Manager. Configure separate profiles for import and export operations.

Configuring the Default Integration Profile by Using adprofileconfig.sh Use this method when:

- The Microsoft Active Directory has no custom schema changes to the objects to be synchronized—that is, the user and group object attributes and object classes are the default ones
- No custom schema elements have been added to the user or group object attributes and object classes

At the end of synchronization, user and group objects synchronized from the Microsoft Active Directory are visible to Oracle components integrated with the Oracle Application Server infrastructure.

The script `adprofileconfig.sh` resides in `$`*`ORACLE_HOME`*`/ldap/odi/admin`. It prompts you for the following:

- Oracle Internet Directory super user DN and password
- Microsoft Active Directory URL (*`host`*`:`*`port`*)
- Microsoft Active Directory user account and password to be used by the Active Directory connector
- Active Directory domain to be synchronized

Once you have entered the parameter values, `adprofileconfig.sh` invokes the Directory Integration and Provisioning Assistant to set up the Active Directory connection information and mapping rules information in the default Active Directory integration profiles.

Configuring the Default Integration Profile by Using the Directory Integration and Provisioning Assistant or Oracle Directory Manager Use this method when:

- The Microsoft Active Directory default has been customized according to the deployment requirements—that is, the user or group objects have custom schema elements
- Oracle Internet Directory objects have custom schema elements
- The objects and attributes to be synchronized between the two directories are not the default ones

To configure the directory integration profile by using this method, follow these steps:

1. Configure the mapping rules as described in "Configuring Mapping Rules" on page 43-10.
2. Update the default parameters as described in "Updating the Default Parameters" on page 43-11.
3. Bootstrap the directories as described in "Task 3: Prepare Both Directories for Synchronization" on page 43-13. The default mapping file for bootstrapping should be like the `activechgimp.map.master` file. When you make changes, use this file as the sample.
4. Configure password synchronization. The default mapping rules are not appropriate for password synchronization between the Microsoft Active Directory and Oracle Internet Directory.

   If the password needs to be synchronized from Oracle Internet Directory to Microsoft Active Directory, then insert the following mapping rule to the mapping file and upload the mapping file to the profile.

   ```
   Userpassword: : :person:unicodepwd: :user
   ```

   Note that the password synchronization from Oracle Internet Directory to Microsoft Active Directory can be carried out only in SSL mode 2—that is, server-only authentication.

   > **See Also:** "Managing the SSL Certificates of Oracle Internet Directory and Connected Directories" on page 35-8 for information about how to store the certificates for running in SSL mode 2

Configuring Mapping Rules

The default profiles have the default mapping rules for mapping the user and group attributes and object classes in Microsoft Active Directory to those on Oracle Internet Directory. These mapping rules assume that no user- and group-specific schema changes have been made to either directory after installation. If there are such changes, then they must be appropriately reflected in the mapping files.

To verify and modify the mapping rules, do the following:

1. Decide which domains, or containers, you want to synchronize. In the case of Microsoft Active Directory deployments with a single domain controller, the container to be specified for synchronization is normally the default naming

context of Microsoft Active Directory. Alternatively, that container can be any subtree under the default naming context.

To identify the default naming context of an AD installation, enter this command:

```
ldapsearch -h Active_Directory_host_name -D bind_DN -w bind_password -b ""
-s base "objectclass=*" defaultnamingcontext
```

2. Decide on the objects—that is, the types of entries—to be synchronized. In an identity management environment these are typically user and group entries.
3. Identify the attributes and how you want to map them between the directories during synchronization.
4. Generate a mapping file with appropriate mapping rules.

> **See Also:** "Format of the Mapping Rules Attribute" on page 33-7 for instructions on how to generate a mapping file

Updating the Default Parameters

Once the mapping file is generated, all the previously-mentioned attributes can be updated by using Oracle Directory Manager or the Directory Integration and Provisioning Assistant. The significance of each attribute is described in Table B–20

on page B-18. Some of those attributes, listed in Table 43–2, have values specific to integration with Microsoft Active Directory.

Table 43–2 Default Attribute Values in the Microsoft Active Directory Integration Profile

Attribute	Value
Profile Name (`orclodipAgentName`)	The default value for the import profile when you are using the DirSync-based approach is `ActiveImport`.
	The default value for the import profile when you are using the USNChange-based approach is `ActiveChgImport`.
	The default value for the export profile is `ActiveExport`.
	This attribute is mandatory.
Connected Directory Account (`orclodipConDirAccessAccount`)	For import cases, the user should have appropriate privileges to use the DirSynchControl or USNBased approach.
	If changes in Oracle Internet Directory are to be exported to Microsoft Active Directory server, then the user must have add and modify privileges to the synchronization domain.
	Note: Create a user account in Microsoft Active Directory exclusively for the Active Directory connector
OID Matching Filter (`orclOdipOIDMatchingFilter`)	orclObjectGUID. In an import operation, this attribute value indicates that the object GUID is the attribute to map entries between Oracle Internet Directory and Microsoft Active Directory.
Agent Execution Command (`orclodipAgentExeCommand`)	This field must be empty.

See Also:

- "Registration of Connectors into the Oracle Directory Integration and Provisioning Platform" on page 33-7 for the required steps
- "The Directory Integration and Provisioning Assistant" on page A-107

Task 2: Configure Access Control Lists

Set up appropriate Access Control Lists (ACLs) allowing read, add, or modify access rights on the subscribed containers.

During import operations, the changes from Microsoft Active Directory are imported to Oracle Internet Directory and made with the identity of the profile with the following DN:

```
orclodipagentname=ActiveImport, cn=subscriber profile, cn=changelog subscriber,
cn=oracle internet directory
```

In this DN, `ActiveImport` is the name of the profile which imports data from Microsoft Active Directory to Oracle Internet Directory. This profile identity needs to be given appropriate privileges to access the containers to which the objects (users or groups) are synchronized. If there are no ACLs applied to the containers of interest, then you can use the following sample LDIF file.

```
dn: container_for_users_and_groups
changetype: modify
add: orclaci
orclaci: access to entry by
"orclodipagentname=ActiveImport,cn=subscriber profile,cn=changelog
subscriber,cn=oracle internet directory" (browse,add,delete)
orclaci: access to attr=(*) by
"orclodipagentname=ActiveImport,cn=subscriber profile,cn=changelog
subscriber,cn=oracle internet directory" (read,search,write,compare)"
```

On the other hand, the privileges can also be granted to the group `cn=odipgroup,cn=odi,cn=oracle internet directory` of which the profile is a member. However, remember that when privileges are granted to the group, all members of the group are, intentionally or not, granted privileges.

Task 3: Prepare Both Directories for Synchronization

Once the profile is created and appropriate ACLs are set, do the following:

1. Make sure that the information of interest to both the directories is equivalent. This requires bootstrapping of data from one directory to the other. This can be achieved by the Directory Integration and Provisioning Assistant with the bootstrap option. Bootstrapping is described in Chapter 37, "Bootstrapping of a Directory in the Oracle Directory Integration and Provisioning Platform".

   Once the bootstrapping is accomplished the profile status attributes are appropriately updated.

2. If you have used LDIF file-based bootstrapping, then you need to initialize the `lastchangenumber` value. This can be done using the Directory Integration and Provisioning Assistant:

   ```
   Dipassistant mp -updlcn
   ```

3. If two-way synchronization is required—that is, synchronization from Oracle Internet Directory to Microsoft Active Directory is also required, then enable the

export profile and make sure that the change logging option is enabled for the Oracle directory server. Change logging is controlled by the `-l` option while starting Oracle Internet Directory. By default it is set to `TRUE`, meaning that change logging is enabled. If it is set to `FALSE`, then shut down the Oracle directory server and start with the change log enabled by using the OID Control Utility.

Task 4: (Optional) Configure the Active Directory External Authentication Plug-in

If you are storing passwords in Microsoft Active Directory, then you must use the Active Directory external authentication plug-in to authenticate Microsoft Active Directory users from Oracle Internet Directory.

This section tells how to install, delete, enable, and disable the Active Directory external authentication plug-in.

Installing Active Directory External Authentication Plug-ins

To install the plug-in:

1. Execute `$`*`ORACLE_HOME`*`/ldap/admin/oidspadi.sh`.

   Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

   - Cygwin 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
   - MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

   To execute oidspadi.sh, enter:

   ```
   cd $ORACLE_HOME/ldap/admin
   sh oidspadi.sh
   ```

   If you are using the Windows operating system, then execute oidspadi.sh after you have installed the UNIX emulation utility by entering:

   ```
   sh oidspadi.sh.
   ```

2. Enter the Microsoft Active Directory host name. This is the Microsoft Active Directory to which you are going to synchronize. This value is required.

3. Enter the Microsoft Active Directory port number. In a multiple domain environment, the default port can be that of the global catalog server, namely, 3268.
4. Enter directory server host name. This value is required.
5. Enter directory server port number. The default port is 389.
6. Enter the password of the Oracle administrator (`orcladmin`). This value is required.
7. Enter the distinguished name of the container to which the plug-in needs to be applied. Every entry in this container will be authenticated against Microsoft Active Directory. Note that this need not necessarily be the User Search Base supplied in Oracle Internet Directory Self-Service Console. All the users under this search base are authenticated externally to the Microsoft Active Directory. If more than one container is specified, then separate the DNs with semi-colons (;).
8. Enter the value of the entry that is to be excluded from authentication to Microsoft Active Directory. This value is the exception to Step 7. You need to enter the value in the standard ldapsearch filter format. For example, if you specify the value `(&(objectclass=inetorgperson)(cn=orcladmin))`, then any entry under the user container specified in Step 7 that has the `cn=orcladmin` and `objectclass=inetorgperson` attribute values will not be authenticated to Microsoft Active Directory.
9. Enter the Plug-in Request Group DN. For security reasons, the plug-in can be invoked only by users belonging to this group. For example, suppose that the Oracle Application Server Single Sign-On administrators are in the group `cn=OracleUserSecurityAdmins,cn=Groups,cn=OracleContext`. If you enter this DN as the vale for the Plug-in Request Group DN, then only requests coming from members of the Oracle Application Server Single Sign-On administrators can trigger the external authentication plug-in. You can enter multiple DN values. Use a semicolon (;) to separate them. This value is not required, but, for security purposes, it should be specified.
10. Enter the choice of using SSL connection to Active Directory or not. If you choose to use SSL, then you need to enter the following:
    a. The Active Directory SSL connection port number.
    b. The location of the Oracle wallet. This wallet needs to have the valid certificate from the Active Directory that you are trying to connect to.
    c. The Oracle wallet password.

When specifying the wallet location on the Microsoft Windows operating system, add an additional backslashes (\). For example, if the wallet location is `D:storage\wallet`, then enter `D:\\storage\\wallet`.

11. Specify the backup Microsoft Active Directory domain controller details (optional).

Deleting the Microsoft Active Directory External Authentication Plug-in

To delete the Active Directory external authentication plug-in, use these commands.

```
ldapdelete -h host -p port -D cn=orcladmin -w password
"cn=adwhencompare,cn=plugin,cn=subconfigsubentry"

ldapdelete -h host -p port -D cn=orcladmin -w password
"cn=adwhenbind,cn=plugin,cn=subconfigsubentry"
```

Enabling the Active Directory External Authentication Plug-ins

To enable the Active Directory external authentication plug-ins, use these two commands:

```
ldapmodify -h host  -p port  -D cn=orcladmin -w password  <<EOF
dn: cn=adwhencompare,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 1
EOF


ldapmodify -h host  -p port  -D cn=orcladmin -w password  <<EOF
dn: cn=adwhenbind,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 1
EOF
```

Disabling the Active Directory External Authentication Plug-ins

To disable the Microsoft Active Directory external authentication plug-ins, use these two commands:

```
ldapmodify -h host -p port -D cn=orcladmin -w password <<EOF
dn: cn=adwhencompare,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 0
```

```
EOF

ldapmodify -h host -p port -D cn=orcladmin -w password <<EOF
dn: cn=adwhenbind,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 0
EOF
```

Microsoft Active Directory External Authentication Plug-in Debugging

If you are experiencing unknown errors, then you can enable the plug-in debugging. To do this, enter:

```
sqlplus ods/odspassword @$ORACLE_HOME/ldap/admin/oidspdon.pls
```

To check the plug-in debugging log, enter:

```
sqlplus ods/ods
select * from plg_debug_log order by id;
```

To delete the plug-in debugging log:

```
sqlplus ods/ods
truncate table plg_debug_log
```

To disable the plug-in debugging:

```
sqlplus ods/ods @$ORACLE_HOME/ldap/admin/oidspdof.pls
```

Note:

- If you need to change the plug-in setup—that is, the information you entered in the installation steps—then you can rerun the installation script. Before you rerun the script, delete the Active Directory external authentication plug-in by following the instructions in "Deleting the Microsoft Active Directory External Authentication Plug-in" on page 43-16.
- You can modify the ODS password by using the OID Database Password Utility (oidpasswd) described in "OID Database Password Utility (oidpasswd) Syntax" on page A-131

Task 5: Start the Synchronization

To start synchronization:

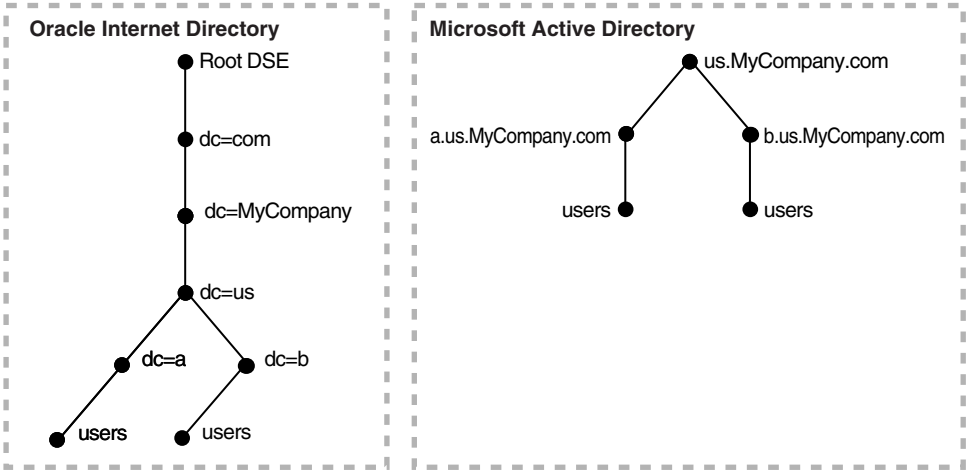
1. Enable the profile by setting the `profileStatus` attribute to ENABLE in either Oracle Directory Manager or the Directory Integration and Provisioning Assistant
2. Start the Oracle directory integration and provisioning server by using the OID Control Utility (oidctl) with the appropriate configuration set entry in which the profile is stored.

Integration of Oracle Internet Directory and a Microsoft Active Directory with Multiple Domains

In this typical scenario, Microsoft Active Directory has multiple domain controllers. A deployment with multiple domain controllers can have one single DIT or a forest of trees. The mapping between the DIT on Oracle Internet Directory and the DIT on Microsoft Active Directory for the single tree and the forest configurations are shown in Figure 43–2 on page 43-18 and Figure 43–3 on page 43-19 respectively.

Figure 43–2 shows an example of how multiple domains in Microsoft Active Directory are mapped to a DIT in Oracle Internet Directory.

Figure 43–2 Integration of Oracle Internet Directory with Multiple Domains in Microsoft Active Directory

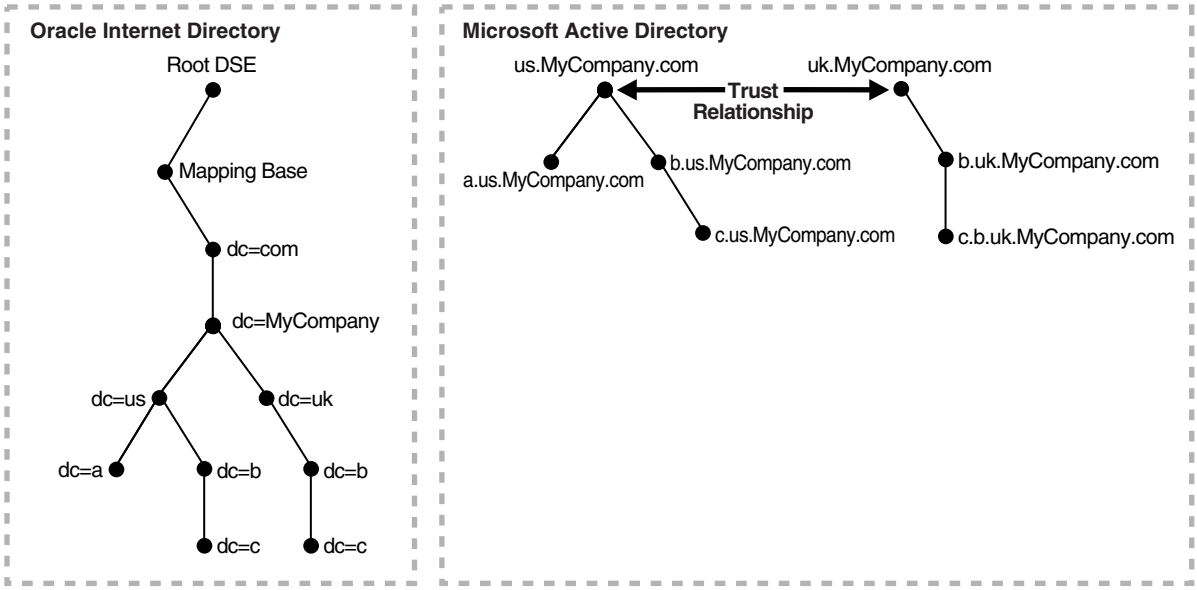


In Figure 43–2, the Microsoft Active Directory environment has a parent and two child domains. Each domain has a domain controller associated with it. The Microsoft Active Directory supporting the node `us.mycompany.com` is the Global Catalog Server.

The first child domain—namely, `a.us.MyCompany.com`—maps to `dc=a,dc=us,dc=MyCompany,dc=com` in Oracle Internet Directory. The second child domain—namely, `b.us.MyCompany.com`, maps to `dc=b,dc=us,dc=MyCompany,dc=com` in Oracle Internet Directory. The common domain component in Microsoft Active Directory environment—namely, `us.MyCompany.com`—maps to the default identity management realm in Oracle Internet Directory—namely, `dc=us,MyCompany,dc=com`.

Figure 43–3 shows how a forest in Microsoft Active Directory is reflected in Oracle Internet Directory.

Figure 43–3 Mapping Between Oracle Internet Directory and a Forest in Microsoft Active Directory



In Figure 43–3, Microsoft Active Directory is the enterprise directory. In this directory, two domain trees constitute a forest, and this forest maps to an identically structured subtree in Oracle Internet Directory. If Oracle Internet Directory serves as a host, then this subtree is located below the base of the identity management

realm for the hosted company. If Oracle Internet Directory is deployed in a non-hosted environment, then it is located just below the root DSE.

The deployment in Figure 43–3 assumes that each user has a single unique identifier as well as group authorizations and profile information for the domain in which the user is provisioned. Trust relationships between the domains enable users in one domain to access resources in another.

Task 1: Configure the Integration Profiles

Based on the method of synchronization, one or more profiles must be configured for importing from multiple domain controllers. If the DirSync approach is followed for synchronization, then a profile must be created to synchronize from each of the domain controllers. If the USNChanged approach is used, then one single profile connecting to the Global Catalog Server can synchronize changes for all the domain controllers. In the connected directory URL, be sure to specify the Global Catalog Server host and port. The default port number is 3268.

Mapping rules need to be configured based on the same approach just discussed. Oracle Corporation recommends that attribute mapping rules be consistent across different domain controllers mapped to Oracle Internet Directory.

Task 2: Configure Access Control Lists

During import operations, changes from Microsoft Active Directory are imported to Oracle Internet Directory and given the following identity:

```
orclodipagentname= profile_name,cn=subscriber profile,
cn=changelog subscriber,cn=oracle internet directory
```

Give the profile identity the appropriate privileges to access the containers in which the objects—that is, the users and groups—are synchronized. For instance, if there are no ACLs applied to the containers of interest, then you can use the following sample LDIF file:

```
dn: container_for_users_and_groups
changetype: modify
add: orclaci
orclaci: access to entry by
"orclodipagentname=profile_name,cn=subscriber profile,cn=changelog
subscriber,cn=oracle internet directory" (browse,add,delete)
orclaci: access to attr=(*) by
"orclodipagentname=profile_name,cn=subscriber profile,cn=changelog
subscriber,cn=oracle internet directory" (read,search,write,compare)"
```

On the other hand, the privileges also can be granted to the group `cn=odipgroup,cn=odi,cn=oracle internet directory`. However, remember that, when the privileges are granted to the group, all the members of the group are granted privileges whether it is intended or not.

Task 3: Prepare the Directories for Synchronization

Once the profile is created and appropriate ACLs are set, but before the start of the synchronization,

- Make sure that the information of interest in both the directories is equivalent. This requires bootstrapping data from one directory to the other. You can do this by using the Directory Integration and Provisioning Assistant with the bootstrap option.
- At the end of bootstrapping, if the export profile is enabled, be sure that the change logging option is enabled for the Oracle directory server. By default it is set to `TRUE`, meaning that change logging is enabled. If it is set to `FALSE`, then shut down the Oracle directory server and start with the change log enabled by using the OID Control Utility. This is described in "The OID Control Utility (oidctl) Syntax" on page A-6.

Task 4: (Optional) Configure the Active Directory External Authentication Plug-in

If you are storing passwords in Microsoft Active Directory, then you must use an Oracle directory server plug-in to authenticate Microsoft Active Directory users from Oracle Internet Directory.

> **See Also:** Task 4: (Optional) Configure the Active Directory External Authentication Plug-in on page 43-14 for the necessary instructions

Task 5: Start the Synchronization

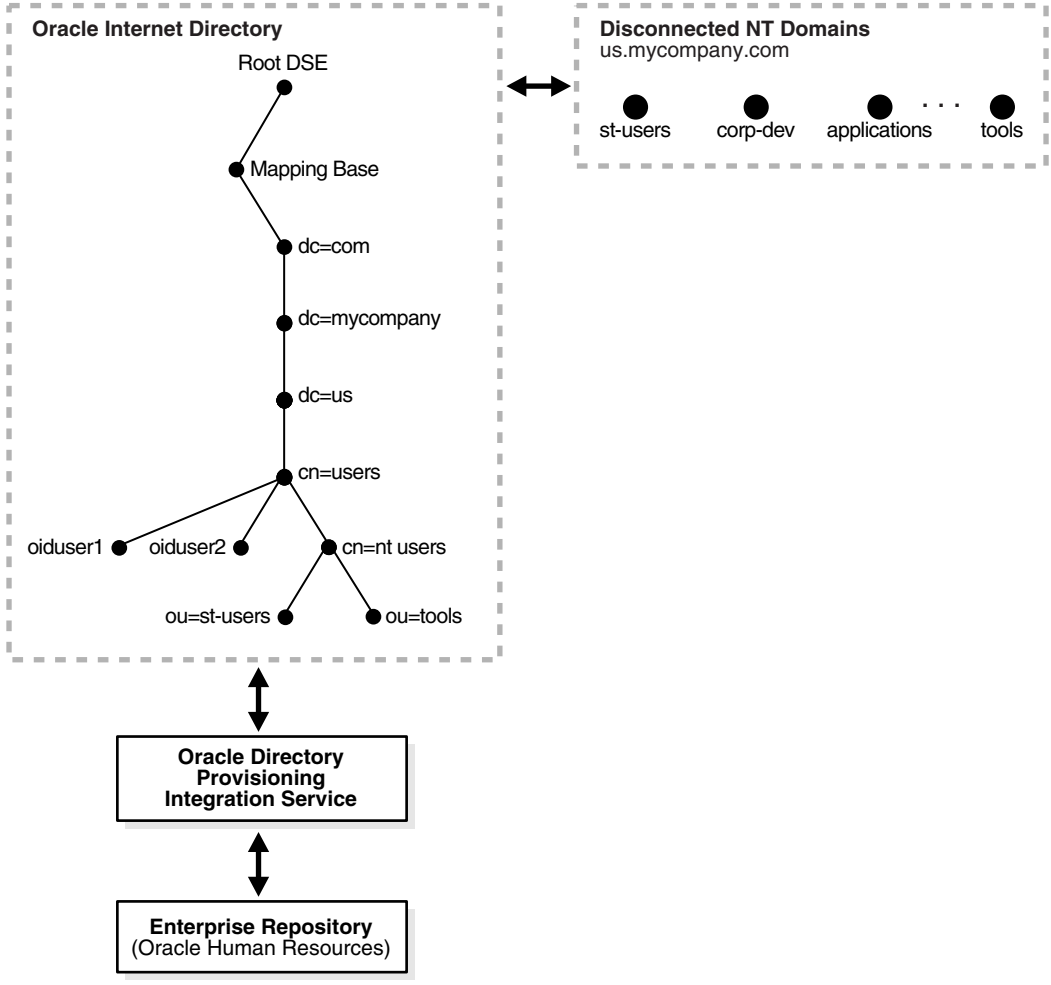
To start the synchronization:

1. Enable the profile by setting the profile status to `ENABLE` in Oracle Directory Manager or the Directory Integration and Provisioning Assistant.
2. Start the Oracle directory integration and provisioning server by using the OID Control Utility (oidctl) with the appropriate configuration set entry in which the profile is stored.

Integration with Microsoft Windows NT 4.0

Microsoft Windows NT domain users can also be integrated into the environment. Microsoft Windows NT groups are not synchronized to Oracle Internet Directory, nor is information about the members of that group. In this case, each of the Microsoft Windows NT domains can be mapped to a domain object or an organization unit object in Oracle Internet Directory. Typical mapping of Microsoft Windows NT domains to domain containers in the Oracle Internet Directory directory information tree is shown in Figure 43–4 on page 43-23.

Figure 43–4 Integration of Oracle Internet Directory DIT with Microsoft Windows NT Domains



Microsoft Windows NT domains are integrated with Oracle Internet Directory so that a minimal user footprint is automatically created in Oracle Internet Directory.

If a user entry exists in Microsoft Windows NT but not in Oracle Internet Directory, then, when that user tries to log in to use the Oracle Application Server components, the auto-registration plug-in creates a shadow entry with minimal footprint information in Oracle Internet Directory. This entry remains in Oracle Internet Directory for the next time the same user tries to log in.

External authentication, with Microsoft Windows NT acting as the external repository, is supported by the use of plug-ins. Ongoing synchronization with the Microsoft Windows NT environment is not supported.

Installation and Configuration of Windows NT External Authentication and Auto-Provisioning Plug-ins

The SQL script `oidspnti.sql` installs the plug-ins that enable Oracle Internet Directory for external authentication against the Microsoft Windows primary domain controller and auto provisioning.

To install the script:

1. Verify that the Oracle Internet Directory server is running.
2. Run the script by entering the following command:

   ```
   cd $ORACLE_HOME/ldap/admin
   sh oidpnti.sh
   ```

3. Enter the Oracle Internet Directory host name and port number. The default port number is 389.
4. Enter the password of the Oracle administrator (`orcladmin`), the directory super user.
5. Enter the distinguished name of the container to which the plug-in needs to be applied. Every entry in this container is then authenticated against the Microsoft Windows NT domain. Note that this need not necessarily be the user search base supplied in the Oracle Internet Directory Self-Service Console. All the users under this search base are authenticated externally to the Microsoft Windows NT domain. If more than one value is specified, then use semi-colons (;) to separate them.
6. Enter the plug-in request group DN. For security reasons, the plug-in can be invoked only by users belonging to this group. For example, suppose that the Oracle Application Server Single Sign-On administrators are in the group `cn=OracleUserSecurityAdmins,cn=Groups,cn=OracleContext`. If you enter this value for the plug-in request group DN, then only the requests coming from Oracle Application Server Single Sign-On administrators can trigger the external authentication plug-in. You can enter multiple DN values. Use a semicolon (;) to separate them. This value is not required, but, for security purposes, should be specified.

7. Choose Auto Registration. The default is Yes. Upon registration, each entry is assigned the object class `orclNTUser`.

At the completion of these steps, the plug-in is installed and enabled.

Enabling the Windows NT External Authentication Plug-in

To enable external authentication, enter these two commands:

```
ldapmodify -h host -p port -D cn=orcladmin -w password <<EOF
dn: cn=ntwhencompare,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 1
EOF

ldapmodify -h host -p port -D cn=orcladmin -w password <<EOF
dn: cn=ntwhenbind,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 1
EOF
```

Disabling the Windows NT External Authentication Plug-in

To disable the external authentication plug-ins, set the value of the attribute `orclpluginenable` to 0 in each of the preceding command.

Enabling Auto Provisioning

To enable auto provisioning, enter the following command:

```
ldapmodify -h host -p port -D cn=orcladmin -w password <<EOF
dn: cn=ntpostsearch,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 1
EOF
```

Disabling Auto Provisioning

To disable auto provisioning, set the value of the attribute `orclpluginenable` to 0 in the preceding command.

Removing External Authentication and Auto Provisioning Plug-ins

To remove external authentication and auto-registration, delete the two plug-in entries from Oracle Internet Directory:

```
ldapdelete -h host -p port D cn=orcladmin -w password
"cn=ntwhencompare,cn=plugin,cn=subconfigsubentry"

ldapdelete -h host -p port -D cn=orcladmin -w password
"cn=ntwhenbind,cn=plugin,cn=subconfigsubentry"

ldapdelete -h host -p port -D cn=orcladmin -w password
"cn=ntpostsearch,cn=plugin,cn=subconfigsubentry"
```

Windows NT External Authentication Plug-in Debugging

If you are experiencing unknown errors, then you can enable the plug-in debugging. To do this:

```
sqlplus ods/odspassword @$ORACLE_HOME/ldap/admin/oidspdon.pls
```

To check the plug-in debugging log:

```
sqlplus ods/ods
select * from plg_debug_log order by id;
```

To delete the plug-in debugging log:

```
sqlplus ods/ods
truncate table plg_debug_log
```

To disable the plug-in debugging:

```
sqlplus ods/ods @$ORACLE_HOME/ldap/admin/oidspdof.pls
```

Note: If you need to change the NT plug-in setup—that is, the information you entered in the installation steps—then you can rerun the installation script. Before you rerun the script, remove the NT external authentication plug-ins by following the preceding instructions.

Limitations of Integration with Microsoft Windows NT Environments in Oracle Internet Directory 10*g* (9.0.4)

To integrate with the Microsoft Windows NT environment, the Oracle Internet Directory should run on the Windows NT operating system.

Sample Integration Profiles and Mapping Rules

The sample integration profiles are created as part of installation by using the Directory Integration and Provisioning Assistant.

The properties file used for creating the profile is located in the $*ORACLE_HOME*/ldap/odi/conf directory.

The sample map file is located in the $*ORACLE_HOME*/ldap/odi/conf directory with the extension of map.master for the various profiles.

The various mapping rules in the files are set for the default configuration of Oracle Internet Directory and Microsoft Active Directory, assuming no schema changes have been made to the Oracle Internet Directory and Microsoft Active Directory domains.

```
DomainRules
#USERBASE refers to the container from which the Microsoft Active Directory
users/groups need to be mapped
# users in Oracle Internet Directory. Typically, it contains the DN of the
domain controller or the users container under the
# domain controller DN.
%USERBASE%:%USERBASE%:
AttributeRules
# USER ENTRY MAPPING RULES
# attribute rule for  mapping the containers between Microsoft Active Directory
and Oracle Internet Directory
#  If you are not wanting to create footprint of the containers, these rules can
be removed
name: : :organizationalunit:ou: : organizationalunit
name: : :container: cn: :orclContainer
name: : :domain: dc: :domain
# Mainatain the Microsoft Active Directory SAMACcount name in a specific Oracle
attribute. This can be used
# for any purpose. Note that it is a mandatory attribute. Invariably all the
Microsoft Active Directory accounts
# have this attribute in their entry.
sAMAccountName:1: :user:orclADSAMAccountName: :orclADUser
# attribute rule for mapping Microsoft Active Directory LOGIN id
```

```
 # Typically, userprincipalname is the Microsoft Active Directory attribute
which is used for login purposes.
# for some of the default entries like 'guest' which needs to be explicitely
enabled with a loginid
# user principalname may not be present. In the default configuration, 'uid' on
Oracle Internet Directory side is chosen
# as the login id attribute.
userPrincipalName: : :user:orclADUserPrincipalName:
:orclADUser:name|userPrincipalName
userPrincipalName: : :user:uid: :inetorgperson:name|userPrincipalName
# attribute rule for mapping entry and to create orclUserV2
# There should be a mapping rule with orcluserv2 objectclass
# without which the PORTAL may not function properly
SAMAccountName:1: :person:sn: : orclUserV2
# attributes to map to cn - normally this is the given name
#name: : :person:displayname: :inetorgperson
name: : :person:cn: :orclUser
givenName: : :person:displayName: :orclUserV2
# mail needs to be assigned valid value for default settings ing DAS
userPrincipalName: : :user:mail: :inetorgperson
ObjectGUID:6: :user:orclADObjectGUID: :orclADUser:bin2b64(ObjectGUID)
ObjectSID:4: :user:orclADObjectSID: :orclADUser:bin2b64(ObjectSID)
# GROUP ENTRY MAPPING RULES
name: : :group:cn: :groupofuniquenames
# displayname needs to be assigned a valid value for default settings on DAS
SAMAccountName: : :group:displayName: :orclgroup
# Description needs tobe assigned a valid value for default settings on DAS
Description: : :group:Description: :orclgroup
member: : :group:uniquemember: :groupofUniqueNames
managedby: : :group:owner: :orclprivilegegroup
sAMAccountName: : :group:cn: :orclgroup
sAMAccountName: : :group:orclADSAMAccountName: :orclADGroup
ObjectGUID:3: :group:orclADObjectGUID: :orclADGroup:bin2b64(ObjectGUID)
ObjectSID:5: :group:orclADObjectSID: :orclADGroup:bin2b64(ObjectSID)
```

Troubleshooting Synchronization with Active Directory Server

You can debug the Active Directory connector by using the oditest utility.

- To troubleshoot the Active Directory import connector, run oditest with `AgentName` as `ActiveChgImp` and look at the `ActiveChgImp.trc` and `ActiveChgImp.aud` files.

- To troubleshoot the default Active Directory connector, run oditest with `AgentName` as `ActiveExport` and look at the `ActiveExport.trc` and `ActiveExport.aud` files.

  See Also: "Troubleshooting Synchronization in the Oracle Directory Integration and Provisioning Platform" on page 33-23 for instructions on using the oditest utility

44

Synchronization with Third-Party Metadirectory Solutions

The Oracle directory integration server does not provide mapping or scheduling services for third-party metadirectory solutions. Instead, 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 third-party metadirectory solutions to 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 third-party metadirectory solution retrieves changes from the change log container and applies them to the third-party directory. To retrieve these changes, the third-party metadirectory solution must subscribe to the Oracle Internet Directory change logs.

Each entry in the change log store has a change number. The third-party metadirectory solution 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 a third-party metadirectory solution retrieved had a number of 250, then subsequent changes it retrieves would have numbers greater than 250.

Note: If a third-party metadirectory solution 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 third-party metadirectory solution 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 33-2 for a conceptual discussion of directory integration profiles

Enabling Third-Party Metadirectory Solutions to Synchronize with Oracle Internet Directory

To enable third-party metadirectory solutions 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 Third-Party Metadirectory Solution

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-53 for instructions on using ldifwrite

Task 2: Create a Change Subscription Object in Oracle Internet Directory for the Third-Party Metadirectory Solution

To enable a third-party metadirectory solution to synchronize with Oracle Internet Directory, you must create a change subscription object for it in Oracle Internet Directory. This gives the third-party metadirectory solution 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 a third-party metadirectory solution 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.

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 44-6 for instructions on temporarily disabling change subscription objects or deleting them altogether

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 Third-Party Metadirectory Solution" on page 44-3.

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 in Multimaster Replication" on page 22-7 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 a third-party metadirectory solution, but you want to disable it temporarily, then set the `orclSubscriberDisable` 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"
```

Part VIII

Directory Plug-ins

This part contains this chapter:

45

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
- Registering and Managing Plug-ins by Using Oracle Directory Manager

> **See Also:** The chapter on the Oracle Internet Directory server plug-in framework in *Oracle Internet Directory Application Developer's Guide*.

About Directory Server Plug-ins

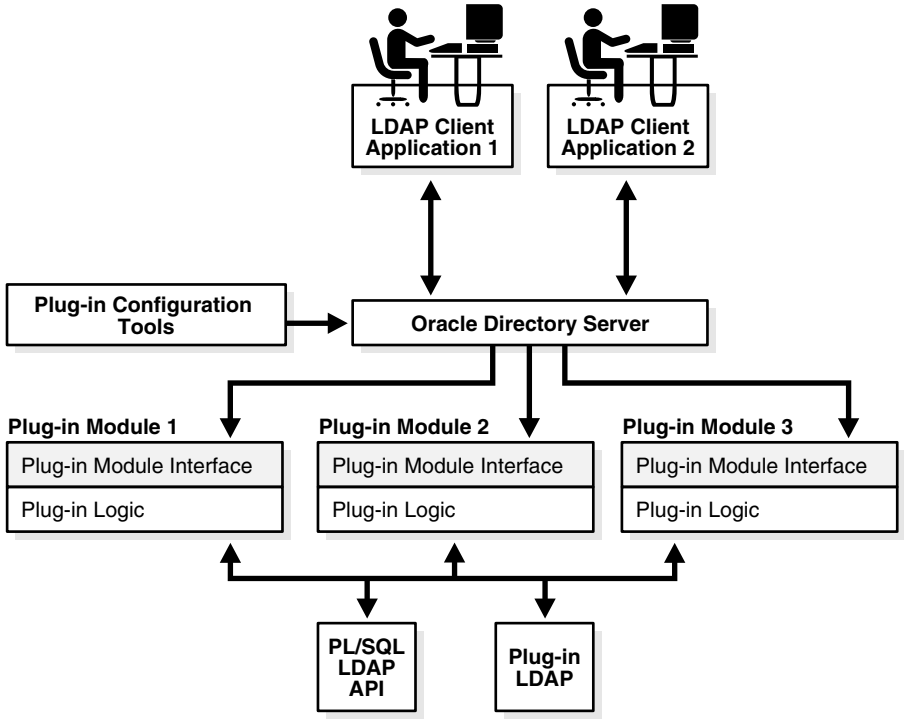
Directory server plug-ins can provide the directory server with the following kinds of added functionality, to mention just a few:

- 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.

In Figure 45–1 on page 45-3, LDAP clients, each using a separate application, send information to and receive it from the Oracle directory server. Plug-in configuration tools likewise send information to the directory server. The directory server sends data to Plug-in Module 1, Plug-in Module 2, and Plug-in Module 3. Each plug-in module has both a plug-in module interface and plug-in logic. Each plug-in module sends information to and receives it from the PL/SQL LDAP API and the Plug-in LDAP.

Figure 45–1 Oracle Internet Directory Plug-in Framework



The work that plug-ins perform depends on whether they execute before, after, or in addition to normal directory server operations. Table 45–1 explains the various kinds of operation-based plug-ins.

Table 45–1 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.

Table 45–1 (Cont.) Types of Operation-Based Plug-ins

Type of Plug-in	Description
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. There are different types of When-operation plug-ins—namely, Add-on and Replace. The Add-on plug-in can perform ldapadd, ldapdelete, and ldapmodify operations. The Replace plug-in can perform ldapcompare, ldapbind, and ldapmodify operations. For example, for the ldapcompare operation, you can use the When Add-on type plug-in. Oracle Internet Directory server executes its server compare code and executes the plug-in module defined by the plug-in developer. For the Replace Type plug-in, Oracle Internet Directory does not execute its own compare code. Instead, it relies on the plug-in module to do the comparison and pass back the compare result. The server comparison procedures are replaced by the plug-in module.

Registering and Managing 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 a configuration entry for the plug-in under `cn=plugin,cn=subconfigsubentry`. This plug-in must have `orclPluginConfig` as one of its object classes.

> **See Also:** "Plug-in Schema Elements" on page B-32 for details about the attributes in the `orclPluginConfig` object class.

This section contains these topics:

- Registering and Managing Plug-ins by Using Oracle Directory Manager
- Registering and Managing Plug-ins by Using Command-Line Tools

Registering and Managing Plug-ins by Using Oracle Directory Manager

This section provides examples of how to create, modify, and delete plug-in configuration entries by using Oracle Directory manager.

Adding a Plug-in Configuration Entry by Using Oracle Directory Manager

To register a plug-in:

1. In the navigator pane, expand in succession **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Plug-in Management**. The Plug-in Management window appears in the right pane.
3. Choose **Create**. The New Plug-in dialog box appears.
4. In the New Plug-in dialog box, enter values in the fields. These fields are described in Table C–13 on page C-10.
5. When you have finished entering the values, choose **OK**. This returns you to the Plug-in Management window. The plug-in you just created is listed in the Plug-in Entry Name column.
6. Choose **OK**.

Editing a Plug-in by Using Oracle Directory Manager

To edit a plug-in entry:

1. In the navigator pane, expand in succession **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Plug-in Management**. The Plug-in Management window appears in the right pane.
3. In the right pane, select the name of the plug-in entry you want to edit, then choose **Edit**. The Plug-in: dialog box appears.
4. In the Plug-in: dialog box, modify the values in the appropriate fields. These fields are described in Table C–13 on page C-10.
5. Choose **OK**.

Deleting a Plug-in by Using Oracle Directory Manager

To delete a plug-in:

1. In the navigator pane, expand in succession **Oracle Internet Directory Servers** and ***directory server instance***.
2. Select **Plug-in Management**. The Plug-in Management window appears in the right pane.
3. In the right pane, select the name of the plug-in you want to delete, then choose **Edit**. The Plug-in: dialog box appears.
4. In the Plug-in dialog box, choose **Delete**, and, when prompted, confirm your deletion. This returns you to the Plug-in Management window. The plug-in entry you deleted no longer appears in the list.

Registering and Managing Plug-ins by Using Command-Line Tools

This section provides examples of how to create, modify, and delete plug-in configuration entries by using command-line tools.

> **See Also:** Plug-in Schema Elements for information about the attributes in the orclPluginConfig object class

Examples: Adding a Plug-in Configuration Entry by Using Command-Line Tools

In the following examples, an entry is created for an operation-based plug-in called `my_plugin1`. The LDIF file is named `my_ldif_file.ldif`.

Example 1: Creating an Operation-Based Plug-in Entry for Compare Operations 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: Creating an Operation-Based Plug-in Entry for Modify Operations The following is an example LDIF file to create such an object:

```
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 timing and the type of LDAP operation related to the plug-in.

Notes: To avoid creating an inconsistent state, metadata for the plug-in configuration entry, `cn=plugin, cn=subconfigsubentry`, is not replicated in the replication environment.

Example: Modifying a Plug-in Configuration Entry by Using Command-Line Tools

This is an example of disabling a plug-in:

```
ldapmodify -h  host_name -p  port_number -D cn=orcladmin -w orcladminpwd <<EOF
dn: cn=post_mod_plugin,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclPluginEnable
orclPluginEnable: 0
EOF
```

Example: Deleting a Plug-in Configuration Entry by Using Command-Line Tools

This is an example of deleting a plug-in:

```
ldapdelete -h  host_name -p port_number -D cn=orcladmin -w orcladminpwd
"cn=post_mod_plugin,cn=plugin,cn=subconfigsubentry"
```

46

Oracle Internet Directory Plug-In for Password Policies

Oracle Internet Directory uses plug-ins to add password value checking to its other password policy management capabilities. These plug-ins enable you to verify that, for example, a new or modified password has the specified minimum length. You can customize password value checking to meet your own requirements.

This chapter contains these topics:

How the Password Policy Plug-in Works

When a user wants to add or modify a password, customized password value checking takes place as follows:

1. The client sends the directory server either an ldapadd or ldapmodify request.
2. Before the directory server makes the addition or modification, it passes the password value to the plug-in.
3. The plug-in
   a. Parses the entry
   b. Captures the `userpassword` attribute value in clear text
   c. Implements whatever password value checking you have specified
4. If the password meets the specification, then the plug-in notifies the directory server accordingly, and the directory server makes the addition or modification. Otherwise, the plug-in sends one of the following error messages to the directory server, which, in turn, passes it to the client.

```
ldap_add: UnKnown Error Encountered
ldap_add: additional info: PASSWORD POLICY VIOLATION:0000X, less than 8
chars

ldap_add: UnKnown Error Encountered
ldap_add: additional info: PASSWORD POLICY VIOLATION:0000X, contains
dictionary word
```

   The same logic applies to the PRE ldapmodify plug-in.

The various kinds of value checks that the password policy plug-in can perform are:

- Minimum and maximum number of alphabetic characters
- Maximum number of numeric characters
- Minimum and maximum number of punctuation characters
- Maximum number of consecutive characters
- Maximum number of instances of any character

Example: Installing, Configuring, and Enabling a Customized Password Policy Plug-in

This example uses the a PL/SQL program, `pluginpkg.sql`, which is described in "Contents of Sample PL/SQL Package pluginpkg.sql" on page 46-5. In general, this package contains:

- Two plug-in modules—`pre_add` and `pre_modify`
- One value checking function, `isGoodPwd`, which verifies that a password meets the minimum length requirement of eight characters

Thus, in this example, if you try to add a user with the `userpassword` value less than eight characters, then the request is rejected. Similarly, if you try to modify a user password, and the new password value is less than eight characters, then the request is rejected.

This section contains these topics:

Loading and Registering the PL/SQL Program

Having implemented the standalone value checking PL/SQL program, do the following:.

1. Load the plug-in package into the database. In this example, we enter:

```
sqlplus ods/odspwd @pluginpkg.sql
```

2. Register the plug-in. This example uses a file named `pluginreg.dat`, which contains the following:

```
### add plugin ###
dn: cn=pre_add_plugin,cn=plugin,cn=subconfigsubentry
objectclass:orclPluginConfig
objectclass:top
orclpluginname:pwd_plugin
orclplugintype:operational
orclplugintiming:pre
orclpluginldapoperation:ldapadd
```

```
orclpluginenable:1
orclpluginversion:1.0.1
cn:pre_add_plugin
orclpluginsubscriberdnlist:dc=com;o=IMC ,c=US
orclpluginattributelist:userpassword

### modify plugin ###
dn: cn=pre_mod_plugin,cn=plugin,cn=subconfigsubentry
objectclass:orclPluginConfig
objectclass:top
orclpluginname:pwd_plugin
orclplugintype:operational
orclplugintiming:pre
orclpluginldapoperation:ldapmodify
orclpluginenable:1
orclpluginversion:1.0.1
cn:pre_mod_plugin
orclpluginsubscriberdnlist:dc=com;o=IMC ,c=US
orclpluginattributelist:userpassword
```

Note that, in this plug-in, we let the directory server know that there are two plug-in modules to invoke when it receives ldapadd or ldapmodify requests. We use `orclpluginsubscriberdnlist:dc=com;o=IMC,c=US` so that the plug-in is invoked ONLY if the target entry is under `dc=com` or `o=IMC,c=US`.

To add this file to the directory, enter the following:

```
ldapadd -p portnum -h hostname -D cn=orcladmin -w orcladminpwd -v -f
pluginreg.dat
```

Coding the Password Policy Plug-in

You can use standard PL/SQL character functions to process the password value. Download any PL/SQL program that can do regular expression. The important thing is to integrate the value checking functions with your plug-in modules.

Debugging the Password Policy Plug-in

Turn on the directory server plug-in to help you examine the process and content of plug-ins.

To setup the directory server plug-in debugging, execute the following command:

```
sqlplus ods/password @$ORACLE/ldap/admin/oidspdsu.pls
```

To enable directory server plug-in debugging, execute the following command:

```
sqlplus ods/password @$ORACLE/ldap/admin/oidspdon.pls
```

To disable directory server plug-in debugging, execute the following command:

```
sqlplus ods/password @$ORACLE/ldap/admin/oidspdof.pls
```

To show directory server plug-in debugging messages, execute the following command:

```
sqlplus ods/password @$ORACLE/ldap/admin/oidspdsh.pls
```

To delete directory server plug-in debugging messages, execute the following command:

```
sqlplus ods/password @$ORACLE/ldap/admin/oidspdde.pls
```

Contents of Sample PL/SQL Package pluginpkg.sql

The script `pluginpkg.sql`, as used in this example, contains the following:

```
CREATE OR REPLACE PACKAGE pwd_plugin AS

PROCEDURE pre_add (ldapplugincontext IN  ODS.plugincontext,
   dn       IN  VARCHAR2,
   entry    IN  ODS.entryobj,
   rc       OUT INTEGER,
   errormsg OUT VARCHAR2
   );

PROCEDURE pre_modify (ldapplugincontext IN  ODS.plugincontext,
      dn       IN  VARCHAR2,
      mods     IN  ODS.modlist,
      rc       OUT INTEGER,
      errormsg OUT VARCHAR2
      );

-- Function: isGoodPwd
-- Parameter: inpwd
-- Purpose: simple password validation function
--          if the password is less than 8 chars
--          this function will return 0, indicating that
--          it is not a good password
```

```
FUNCTION isGoodPwd(inpwd IN VARCHAR2)
  RETURN INTEGER;


END pwd_plugin;
/

show error

CREATE OR REPLACE PACKAGE BODY pwd_plugin AS

FUNCTION isGoodPwd(inpwd IN VARCHAR2)
  RETURN INTEGER
  IS
     i NUMBER;
     ret NUMBER DEFAULT 1;
     minpwdlen NUMBER DEFAULT 8;
     len       NUMBER DEFAULT 0;
BEGIN
   plg_debug( '=== begin of ISGOODPWD ===');
   plg_debug( 'password = ' || inpwd);
   len := LENGTH(inpwd);
   plg_debug( 'password length = ' || len);

   IF len < minpwdlen THEN
      RETURN 0;
    ELSE
      RETURN ret;
   END IF;

   plg_debug( '=== end of ISGOODPWD ===');

EXCEPTION
   WHEN OTHERS THEN
      plg_debug( 'Exception in isGoodPwd(). Error code is ' || TO_
CHAR(SQLCODE));
      plg_debug( '   ' || Sqlerrm);
      RETURN 0;
END;


PROCEDURE pre_add (ldapplugincontext IN  ODS.plugincontext,
   dn       IN  VARCHAR2,
   entry    IN  ODS.entryobj,
```

```
   rc       OUT INTEGER,
   errormsg OUT VARCHAR2
   )
  IS
     inpwd VARCHAR2(256) DEFAULT NULL;
     ret   NUMBER        DEFAULT 1;
BEGIN
   plg_debug( '=== begin of PRE_ADD_PLUGIN ===');
   plg_debug( 'dn = ' || dn);

   plg_debug( 'entry obj ' || ':entryname = ' || entry.entryname);

   FOR l_counter1 IN 1..entry.attr.COUNT LOOP
      plg_debug( 'attrname[' || l_counter1 || '] = ' ||
 entry.attr(l_counter1).attrname);
      FOR l_counter2 IN 1..entry.attr(l_counter1).attrval.COUNT LOOP
 plg_debug( entry.attr(l_counter1).attrname ||
    '[' || l_counter1 || ']' ||
    '.val[' || l_counter2 || '] = ' ||
    entry.attr(l_counter1).attrval(l_counter2));
      END LOOP;

      IF entry.attr(l_counter1).attrname = 'userpassword' THEN
 inpwd := entry.attr(l_counter1).attrval(1);
 -- assuming only one attr val for userpassword
      END IF;

  END LOOP;

  IF (inpwd IS NOT NULL) THEN
     ret := isGoodPwd(inpwd);
  END IF;

  IF (inpwd IS NULL OR ret = 0) THEN
     rc := 1;
     errormsg := 'PASSWORD POLICY VIOLATION:0000X, less than 8 chars';
     plg_debug( ' we got an invalid password ');
   ELSE
     plg_debug( ' we got a good password ');
     rc := 0;
     errormsg := 'no pre_mod plguin error msg';
  END IF;

  plg_debug( '=== end of PRE_ADD_PLUGIN ===');
```

```
EXCEPTION
   WHEN OTHERS THEN
      plg_debug( 'Exception in PRE_ADD plugin. Error code is ' || TO_
CHAR(SQLCODE));
      plg_debug( '   ' || Sqlerrm);
      rc := 1;
      errormsg := 'exception: pre_add plguin';
END;

PROCEDURE pre_modify (ldapplugincontext IN  ODS.plugincontext,
      dn       IN  VARCHAR2,
      mods     IN  ODS.modlist,
      rc       OUT INTEGER,
      errormsg OUT VARCHAR2
      )
  IS
     old_passwd VARCHAR2(256) DEFAULT NULL;
     new_passwd VARCHAR2(256) DEFAULT NULL;
     ret        NUMBER        DEFAULT 1;

BEGIN
   plg_debug( '=== begin of PRE_MOD_PLUGIN ===');
   plg_debug( dn);

   FOR l_counter1 IN 1..mods.COUNT LOOP
      IF (mods(l_counter1).operation = 2) AND(mods(l_counter1).type
'userpassword') THEN

 FOR l_counter2 IN 1..mods(l_counter1).vals.COUNT LOOP
    new_passwd := mods(l_counter1).vals(l_counter2).val;
 END LOOP;
      END IF;

      IF (mods(l_counter1).operation = 0) AND
(mods(l_counter1).type = 'userpassword') THEN

 FOR l_counter2 IN 1..mods(l_counter1).vals.COUNT LOOP
    new_passwd := mods(l_counter1).vals(l_counter2).val;
 END LOOP;
      END IF;

      IF (mods(l_counter1).operation = 1) AND
(mods(l_counter1).type = 'userpassword') THEN

 FOR l_counter2 IN 1..mods(l_counter1).vals.COUNT LOOP
```

```
    old_passwd := mods(l_counter1).vals(l_counter2).val;
 END LOOP;
      END IF;
   END LOOP;

   plg_debug(' new password: ' || new_passwd);
   plg_debug(' old password: ' || old_passwd);

   IF (new_passwd IS NOT NULL) THEN
      ret := isGoodPwd(new_passwd);
   END IF;

   IF (new_passwd IS NULL OR ret = 0) THEN
      rc := 1;
      errormsg := 'PASSWORD POLICY VIOLATION:0000X, less than 8 chars';
      plg_debug( ' we got an invalid password ');
    ELSE
      plg_debug( ' we got a good password ');
      rc := 0;
      errormsg := 'no pre_mod plguin error msg';
   END IF;

   plg_debug( '=== end of PRE_MOD_PLUGIN ===');

EXCEPTION
   WHEN OTHERS THEN
      plg_debug( 'Exception in PRE_MODIFY plugin. Error code is ' || TO_
CHAR(SQLCODE));
      plg_debug( '   ' || Sqlerrm);
      rc := 1;
      errormsg := 'exception: pre_mod plguin';
END;

END pwd_plugin;
/
show error

GRANT EXECUTE ON pwd_plugin TO ods_server;

EXIT;
```

47

Setting Up the Customized External Authentication Plug-in

You can store user security credentials in a repository other than Oracle Internet Directory—for example, a database or another LDAP directory—and use these credentials for user authentication to Oracle components. You do not need to store the credentials in Oracle Internet Directory and then worry about keeping them synchronized. Authenticating a user by way of credentials stored in an external repository is called external authentication.

This chapter contains these topics:

Native Authentication Contrasted with External Authentication

Authentication that relies on security credentials stored in Oracle Internet Directory is called native authentication. When a user enters her security credentials, the directory server compares them with the credentials stored in Oracle Internet Directory. If the credentials match, then the directory server authenticates the user.

Authentication that relies on security credentials stored in a directory other than Oracle Internet Directory is called external authentication. When a user enters her security credentials, the directory server compares them with the credentials stored in the other directory. This is done by using:

- A PL/SQL program that does the external authentication work
- An external authentication plug-in that invokes this PL/SQL program

Example: Installing, Configuring, and Enabling the External Authentication Plug-in

This section contains these topics:

Sample PL/SQL Package oidexaup.sql

This example uses the a PL/SQL program, `oidexaup.sql`, which is described in "Contents of PL/SQL Package oidexaup.sql" on page 47-5. This package is used for installing the external authentication plug-in PL/SQL package. It contains two plug-ins—namely, `when_compare_replace` and `when_modify_replace`—and one utility function —namely, `get_nickname`. The integrated package is the plug-in package, `OIDEXTAUTH`. This package can also serve as a template you to modify according to your deployment environment.

To install, configure, and enable the external authentication plug-in, follow these steps:

1. Implement your standalone external authentication PL/SQL program. For example, if you want to authenticate users by using user names and passwords, then you should have a PL/SQL program which takes these two parameters.

In our sample code, `oidexaup.sql`, `auth_external` is the program package name, and `authenticate_user` is the function that does the authentication. You need to make sure that this standalone program is working properly before you move on to next steps.

2. Integrate this standalone program into the plug-in modules.
3. Load the plug-in package into database. In this example, we enter:

```
sqlplus ods/odspwd @oidexaup.sql
```

4. Register the plug-ins. Do this by creating and uploading an LDIF file that provides the directory server with the necessary information to invoke the plug-in.
5. This example uses a file named oidexauth.ldif, which contains the following:

```
dn: cn=whencompare,cn=plugin,cn=subconfigsubentry
objectclass:orclPluginConfig
objectclass:top
orclpluginname:oidextauth
orclplugintype:operational
orclplugintiming:when
orclpluginldapoperation:ldapcompare
orclpluginenable:1
orclpluginversion:1.0.1
orclPluginIsReplace:1
cn:whencompare
orclpluginsubscriberdnlist:dc=com;o=IMC,c=US
orclpluginattributelist:userpassword
orclpluginrequestgroup:$prgdn

dn: cn=whenmodify,cn=plugin,cn=subconfigsubentry
objectclass:orclPluginConfig
objectclass:top
orclpluginname:oidextauth
orclplugintype:operational
orclplugintiming:when
orclpluginldapoperation:ldapmodify
orclpluginenable:1
orclpluginversion:1.0.1
orclPluginIsReplace:1
cn:whenmodify
orclpluginsubscriberdnlist:dc=com;o=IMC,c=US
orclpluginattributelist:userpassword
orclpluginrequestgroup:$prgdn
```

In this file, we notify the directory server that, whenever there is an ldapcompare or ldapmodify request, there are two plug-ins to be invoked.

We use `orclpluginsubscriberdnlist:dc=com;o=IMC,c=US` so that plug-ins will ONLY be invoked if the target entry is under `dc=com` or `o=IMC,c=US`.

Replace `$prgdn` with the plug-in request group DN. This is an optional, recommended security feature. For integrating with Oracle Application Server Single Sign-On, this value is a required field. Only members of the group entered can invoke the plug-ins. You may enter multiple groups. Use a semicolon to separate entries.

The recommended defaults are: `cn=OracleUserSecurityAdmins,cn=Groups,cn=OracleContext` and `cn=OracleDASAdminGroup,cn=Groups,cn=OracleContext,o=default_subscriber,dc=com`. Note that the Oracle Application Server Single Sign-On server is a member of the first group. Also, be sure to replace `o=default_subscriber` with the correct value for your deployment environment.

To add this file to the directory, enter the following:

```
ldapadd -p portnum -h hostname -D cn=orcladmin -w orcladminpwd -v -f
oidexauth.ldif
```

Now, everything should be ready. Use the ldapcompare command-line tool to verify that the plug-in and authentication program are working properly before you try to authenticate user from Oracle Application Server Single Sign-On.

In our example, we also provide the plug-in code for externally modifying user password.

Debugging the External Authentication Plug-in

Turn on directory server plug-in to help you to examine the process and content of plug-ins.

To setup directory server plug-in debugging, execute the following command:

```
sqlplus ods/password @$ORACLE/ldap/admin/oidspdsu.sql
```

To enable directory server plug-in debugging, execute the following command:

```
sqlplus ods/password @$ORACLE/ldap/admin/oidspdon.sql
```

To disable directory server plug-in debugging, execute the following command:

```
sqlplus ods/password @$ORACLE/ldap/admin/oidspdof.sql
```

To show directory server plug-in debugging messages, execute the following command:

```
sqlplus ods/password @$ORACLE/ldap/admin/oidspdsh.sql
```

To delete directory server plug-in debugging messages, please execute the following command:

```
sqlplus ods/password @$ORACLE/ldap/admin/oidspdde.sql
```

Contents of PL/SQL Package oidexaup.sql

The script oidexaup.sql, as used in this example, contains the following:

```
CREATE OR REPLACE PACKAGE OIDEXTAUTH AS

  PROCEDURE when_compare_replace (ldapplugincontext IN  ODS.plugincontext,
                                  result            OUT INTEGER,
                                  dn                IN  VARCHAR2,
                                  attrname          IN  VARCHAR2,
                                  attrval           IN  VARCHAR2,
                                  rc                OUT INTEGER,
                                  errormsg          OUT VARCHAR2
                                  );

  PROCEDURE when_modify_replace (ldapplugincontext IN  ODS.plugincontext,
                                 dn                IN  VARCHAR2,
                                 mods              IN  ODS.modlist,
                                 rc                OUT INTEGER,
                                 errormsg          OUT VARCHAR2
                                 );

  FUNCTION get_nickname (dn         IN  VARCHAR2,
                         my_session IN  DBMS_LDAP.session)
    RETURN VARCHAR2;

END OIDEXTAUTH;
/

SHOW ERROR
```

```
CREATE OR REPLACE PACKAGE BODY OIDEXTAUTH AS

  -- We use this function to convert the dn to nickname.
  -- When OID server receives the ldapcompare request, it
  -- only has the dn information. We need to use DBMS_LDAP_UTL
  -- package to find out the nickname attribute value of
  -- the entry.

  FUNCTION get_nickname (dn         IN VARCHAR2,
                         my_session IN DBMS_LDAP.session)
    RETURN VARCHAR2
    IS
       my_pset_coll        DBMS_LDAP_UTL.PROPERTY_SET_COLLECTION;
       my_property_names   DBMS_LDAP.STRING_COLLECTION;
       my_property_values  DBMS_LDAP.STRING_COLLECTION;

       user_handle         DBMS_LDAP_UTL.HANDLE;
       user_id             VARCHAR2(2000);
       user_type           PLS_INTEGER;
       user_nickname       VARCHAR2(256) DEFAULT NULL;

       my_attrs            DBMS_LDAP.STRING_COLLECTION;
       retval              PLS_INTEGER;

  BEGIN
     plg_debug( '=== Beginning of get_nickname() === ');
     user_type     := DBMS_LDAP_UTL.TYPE_DN;
     user_id       := dn;

     retval := DBMS_LDAP_UTL.create_user_handle(user_handle, user_type, user_
id);

     plg_debug( 'create_user_handle() Returns ' || To_char(retval));

     retval := DBMS_LDAP_UTL.get_user_properties(my_session,
                                                 user_handle,
                                                 my_attrs,
                                                 DBMS_LDAP_UTL.NICKNAME_
PROPERTY,
                                                 my_pset_coll);

     plg_debug( 'get_user_properties() Returns ' || To_char(retval));

     IF my_pset_coll.COUNT > 0 THEN
        FOR i IN my_pset_coll.first .. my_pset_coll.last LOOP
```

```
          retval := DBMS_LDAP_UTL.get_property_names(my_pset_coll(i),
                                                     my_property_names);
          IF my_property_names.COUNT > 0 THEN
             FOR j IN my_property_names.first .. my_property_names.last LOOP
                retval := DBMS_LDAP_UTL.get_property_values(my_pset_coll(i),
                                                            my_property_
names(j),
                                                            my_property_
values);
                IF my_property_values.COUNT > 0 THEN
                   FOR k IN my_property_values.FIRST..my_property_values.LAST
LOOP
                      user_nickname := my_property_values(k);
                      plg_debug( 'user nickname = ' || user_nickname);
                   END LOOP;
                END IF;
             END LOOP;
          END IF; -- IF my_property_names.count > 0
       END LOOP;
    END IF; -- If my_pset_coll.count > 0

    plg_debug( 'got user_nickname: ' || user_nickname);

    -- Free my_properties
    IF my_pset_coll.count > 0 then
       DBMS_LDAP_UTL.free_propertyset_collection(my_pset_coll);
    END IF;

    DBMS_LDAP_UTL.free_handle(user_handle);

    RETURN user_nickname;

  EXCEPTION
    WHEN OTHERS THEN
       plg_debug('Exception in get_nickname. Error code is ' || to_
char(sqlcode));
       plg_debug('   ' || Sqlerrm);
       RETURN NULL;
  END;


  PROCEDURE when_compare_replace (ldapplugincontext IN  ODS.plugincontext,
                                  result            OUT INTEGER,
                                  dn                IN  VARCHAR2,
                                  attrname          IN  VARCHAR2,
```

```
                                attrval          IN  VARCHAR2,
                                rc               OUT INTEGER,
                                errormsg         OUT VARCHAR2
                                )
  IS
     retval pls_integer;
     lresult BOOLEAN;

     my_session          DBMS_LDAP.session;
     my_property_names   DBMS_LDAP.STRING_COLLECTION;
     my_property_values  DBMS_LDAP.STRING_COLLECTION;
     my_attrs            DBMS_LDAP.STRING_COLLECTION;
     my_pset_coll        DBMS_LDAP_UTL.PROPERTY_SET_COLLECTION;
     user_handle         DBMS_LDAP_UTL.HANDLE;

     user_id             VARCHAR2(2000);
     user_type           PLS_INTEGER;
     user_nickname       VARCHAR2(60);
     remote_dn           VARCHAR2(256);

     i                   PLS_INTEGER;
     j                   PLS_INTEGER;
     k                   PLS_INTEGER;

BEGIN
   plg_debug( '=== Begin of WHEN-COMPARE-REPLACE plug-in');
   plg_debug( 'DN = ' || dn);
   plg_debug( 'Attr = ' || attrname);
   --plg_debug( 'Attrval = ' || attrval);

   DBMS_LDAP.USE_EXCEPTION := FALSE;
   errormsg := 'No error msg';
   rc := 0;

   -- converting dn to nickname
   my_session := LDAP_PLUGIN.init(ldapplugincontext);
   plg_debug( 'ldap_session =' || RAWTOHEX(SUBSTR(my_session,1,8)));

   retval := LDAP_PLUGIN.simple_bind_s(ldapplugincontext, my_session);
   plg_debug( 'simple_bind_res =' || TO_CHAR(retval));

   user_nickname := get_nickname(dn, my_session);
   plg_debug( 'user_nickname =' || user_nickname);

   -- unbind from the directory
```

```
     retval := DBMS_LDAP.unbind_s(my_session);
     plg_debug( 'unbind_res Returns ' || To_char(retval));

     IF (user_nickname IS NULL) THEN
        result := 32;
        errormsg := 'Can''t find the nickname';
        plg_debug( 'Can''t find the nickname');
        RETURN;
     END IF;

     plg_debug( '=== Now go to extauth ');

     BEGIN
        retval := auth_external.authenticate_user(user_nickname, attrval);
        plg_debug( 'auth_external.authenticate_user() returns = ' || 'True');
        result := 6; -- compare result is TRUE
     EXCEPTION
        WHEN OTHERS THEN
           result := 5; -- compare result is FALSE
           plg_debug( 'auth_external.authenticate_user() returns = ' ||
'False');
           RETURN;
     END;

     plg_debug( '=== End of WHEN-COMPARE-REPLACE plug-in');
  EXCEPTION
     WHEN OTHERS THEN
        rc := 1;
        errormsg := 'Exception: when_compare_replace plugin';
        plg_debug( 'EXCEPTION: ' || retval);
        plg_debug('Exception in when_compare. Error code is ' || to_
char(sqlcode));
        plg_debug('   ' || Sqlerrm);
  END;


  PROCEDURE when_modify_replace (ldappluginContext IN  ODS.plugincontext,
                                 dn                IN  VARCHAR2,
                                 mods              IN  ODS.modlist,
                                 rc                OUT INTEGER,
                                 errormsg          OUT VARCHAR2
                                 )
    IS
       retval pls_integer;
       lresult BOOLEAN;
```

```
    my_session          DBMS_LDAP.SESSION;
    my_property_names   DBMS_LDAP.STRING_COLLECTION;
    my_property_values  DBMS_LDAP.STRING_COLLECTION;
    my_attrs            DBMS_LDAP.STRING_COLLECTION;
    my_modval           DBMS_LDAP.BERVAL_COLLECTION;
    my_pset_coll        DBMS_LDAP_UTL.PROPERTY_SET_COLLECTION;
    user_handle         DBMS_LDAP_UTL.HANDLE;

    l_mod_array         RAW(32);
    user_id             VARCHAR2(2000);
    user_type           PLS_INTEGER;
    user_nickname       VARCHAR2(2000);
    old_passwd          VARCHAR2(60) DEFAULT NULL;
    new_passwd          VARCHAR2(60) DEFAULT NULL;
    remote_dn           VARCHAR2(256);

    i                   PLS_INTEGER;
    j                   PLS_INTEGER;
    k                   PLS_INTEGER;

BEGIN
   plg_debug( '=== Begin of WHEN-MODIFY-REPLACE plug-in');
   DBMS_LDAP.USE_EXCEPTION := FALSE;
   user_type     := DBMS_LDAP_UTL.TYPE_DN;
   user_id       := dn;

   -- converting dn to nickname
   my_session := LDAP_PLUGIN.init(ldapplugincontext);
   plg_debug( 'ldap_session =' || RAWTOHEX(SUBSTR(my_session,1,8)));

   retval := LDAP_PLUGIN.simple_bind_s(ldapplugincontext, my_session);
   plg_debug( 'simple_bind_res =' || TO_CHAR(retval));

   user_nickname := get_nickname(dn, my_session);
   plg_debug( 'user_nickname =' || user_nickname);

   -- unbind from the directory
   retval := DBMS_LDAP.unbind_s(my_session);

   FOR l_counter1 IN 1..mods.COUNT LOOP
      IF (mods(l_counter1).operation = 2) AND
        (mods(l_counter1).type = 'userpassword') THEN

         FOR l_counter2 IN 1..mods(l_counter1).vals.COUNT LOOP
```

```
            new_passwd := mods(l_counter1).vals(l_counter2).val;
         END LOOP;
      END IF;

      IF (mods(l_counter1).operation = 0) AND
        (mods(l_counter1).type = 'userpassword') THEN

         FOR l_counter2 IN 1..mods(l_counter1).vals.COUNT LOOP
            new_passwd := mods(l_counter1).vals(l_counter2).val;
         END LOOP;
      END IF;

      IF (mods(l_counter1).operation = 1) AND
        (mods(l_counter1).type = 'userpassword') THEN

         FOR l_counter2 IN 1..mods(l_counter1).vals.COUNT LOOP
            old_passwd := mods(l_counter1).vals(l_counter2).val;
         END LOOP;
      END IF;
   END LOOP;

   IF new_passwd IS NOT NULL AND old_passwd IS NOT NULL THEN
     BEGIN
        auth_external.change_passwd(user_nickname, old_passwd, new_passwd);
     EXCEPTION
        WHEN OTHERS THEN
           rc := 1;
           plg_debug( 'auth_external.change_passwd() raised exception.');
           errormsg := 'auth_external.change_passwd() raised exception.';
           RETURN;
     END;
    ELSIF new_passwd IS NOT NULL AND old_passwd IS NULL THEN
     BEGIN
        auth_external.reset_passwd(user_nickname, new_passwd);
     EXCEPTION
        WHEN OTHERS THEN
           plg_debug( 'auth_external.reset_passwd() raised exception.');
           rc := 1;
           errormsg := 'auth_external.reset_passwd() raised exception.';
           RETURN;
     END;
    ELSE
           rc := 1;
           errormsg := 'PLG_Exception. Not enough info to change passwd.';
   END IF;
```

```
     plg_debug( 'external change password succeed');
     rc := 0;
     errormsg := 'No when_mod_replace plguin error msg';

     retval := DBMS_LDAP.unbind_s(my_session);

     plg_debug( 'End of WHEN-MODIFY-REPLACE');
     --COMMIT;
  EXCEPTION
     WHEN others THEN
        rc := 1;
        errormsg := 'PLG_Exception: when_modify_replace plguin';
        plg_debug('Exception in when_modify. Error code is ' || to_
char(sqlcode));
        plg_debug('   ' || Sqlerrm);
  END;

END OIDEXTAUTH;
/
SHOW ERRORS
--list


GRANT EXECUTE ON OIDEXTAUTH TO ods_server;

EXIT;
```

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_* type:	*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
mail: 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 6-7.

- 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 lvii for a complete list of LDIF formatting rules
- "Using Globalization Support with LDIF Files" on page G-3

Starting, Stopping, Restarting, and Monitoring Oracle Internet Directory Servers

This section tells how to use command-line tools for starting, stopping, restarting, and monitoring Oracle Internet Directory servers. It contains these topics:

- The OID Monitor (oidmon) Syntax
- The OID Control Utility (oidctl) Syntax

The OID Monitor (oidmon) Syntax

Use the OID Monitor to initiate, monitor, and terminate directory 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.

Starting the OID Monitor

Starting OID Monitor restarts any Oracle Internet Directory processes that were previously stopped.

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*.AL32UTF8). The default language set at installation is AMERICAN_AMERICA.
   - PATH. In the PATH environment variable, specify the Oracle LDAP binary—that is, *`ORACLE_HOME`*`/bin`—before the UNIX binary directory.

2. At the system prompt, type:

```
oidmon [connect=connect_string] [host=virtual/host_name][sleep=seconds]
start
```

Table A–1 Arguments for Starting OID Monitor

Argument	Description
connect=*connect_string*	Specifies the connect string for the database to which you want to connect. This is the network service name set in the `tnsnames.ora` file. This argument is optional.
host=*virtual/host_name*	Specifies the virtual host or rack nodes on which to start OID Monitor
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
```

To start OID Monitor on a virtual host:

```
oidmon connect=dbs1 host=virtual_host start
```

Stopping the OID Monitor

Stopping the OID Monitor also stops all other Oracle Internet Directory processes.

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

```
oidmon [connect=connect_string] [host=virtual/host_name] stop
```

Table A–2 Arguments for Stopping OID Monitor

Argument	Description
`connect=`*`connect_string`*	Specifies the connect string for the database to which you want to connect. This is the connect string set in the `tnsnames.ora` file.
`host=`*`virtual/host name`*	Specifies the virtual host or rack nodes on which to start OID Monitor
`stop`	Stops the OID Monitor process

For example:

```
oidmon connect=dbs1 stop
```

Starting and Stopping OID Monitor in a Cold Failover Cluster Configuration

While starting and stopping OID Monitor, use the `host` parameter to specify the virtual host name. The syntax is:

```
oidmon [connect=connect_string] host=virtual_host start|stop
```

> **Note:** If you are going to start Oracle Internet Directory servers on a virtual host, then, when using both OIDMON and OIDCTL, be sure to specify the `host` argument as the virtual host.
>
> If the OID Monitor is started with the `host=host name` argument, and the host name does not match the name of the physical host, then the OID Monitor assumes that the intended host is the logical host. You must use the same host name when using OIDCTL to stop or start any servers, otherwise the OID Monitor does not start or stop the servers.
>
> To determine the physical host name, execute the `uname` command.

The OID Control Utility (oidctl) Syntax

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

> **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 and Stopping an Oracle Directory Server Instance
- Troubleshooting Directory Server Instance Startup
- Starting and Stopping an Oracle Directory Replication Server Instance
- Starting the Oracle Directory Integration and Provisioning Server

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=connect_string server=oidldapd instance=server_instance_number
[configset=configset_number] [host=virtual/host_name][flags=' -p port_number
-work maximum_number_of_worker_threads_per_server -debug debug_level -l change_
logging' -server number_of_server_processes] start
```

Table A–3 Arguments for Starting a Directory Server by Using OIDCTL

Argument	Description
`-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.
`-p` *port_number*	Specifies a port number during server instance startup. The default port number is 389.
`-server` *number_of_server_processes*	Specifies the number of server processes to start on this port

Table A–3 (Cont.) Arguments for Starting a Directory Server by Using OIDCTL

Argument	Description
`-sport`	Specifies the SSL port number during server instance startup. Default port if not set is 636. **See Also:** ■ The information about `orclsslenable` attribute in "Configuration Set Entry Schema Elements" on page B-5 ■ "Configuring SSL Parameters" on page 13-3
`-work` *maximum_number_of_worker_threads_per_server*	Specifies the maximum number of worker threads for this server
`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.
`connect=`*connect_string*	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`.
`host=`*virtual/host_name*	Specifies the virtual host or rack nodes on which to start the directory server
`instance=`*server_instance_number*	Instance number of the server to start. Should be a number between 1 and 1000.
`server=oidldapd`	Type of server to start (valid values are OIDLDAPD and OIDREPLD). This is not case-sensitive.
`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 At the system prompt, type:

```
oidctl connect=connect_string server=oidldapd instance=server_instance_number
stop
```

For example:

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

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=connect_string 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`.

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=connect_string 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
```

Table A–4 Arguments for Starting a Directory Replication Server by Using OIDCTL

Argument	Description
connect=*connect_string*	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 At the system prompt, type:

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

For example:

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

Starting the Oracle Directory Integration and Provisioning Server

The Oracle directory integration and provisioning server executable, `odisrv`, resides in the *$ORACLE_HOME*`/bin` directory.

The way you start the directory integration and provisioning server depends on whether your installation is:

- A typical Oracle Internet Directory installation

  In this case, your installation includes, among other server and client components, the OID Monitor and the OID Control Utility. In such installations, you start and stop the directory integration and provisioning server by using these tools.

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

- An Oracle Directory Integration and Provisioning platform-only installation

In this case, the way you start the directory integration and provisioning server depends on whether you are using the Oracle Directory Integration and Provisioning platform for high availability.

- If you are using Oracle Directory Integration and Provisioning platform for high availability, then Oracle Corporation recommends that you start the directory integration and provisioning server by using the OID Monitor and the OID Control Utility. This requires configuring the `tnsnames.ora` file with the right host and SID to which the OID Monitor must connect.
- If you are *not* using Oracle Directory Integration and Provisioning platform for high availability, then Oracle Corporation recommends that you start the directory integration and provisioning server without using the OID Monitor.

You can start the directory integration and provisioning server in either SSL mode for tighter security, or non-SSL mode. You need to use a connect string to connect to the database.

Note: When the Oracle directory integration and provisioning server is invoked in the default mode, it supports only the Oracle Directory Provisioning Integration Service, and not the Oracle Directory Synchronization Service.

Starting the Oracle Directory Integration and Provisioning Server by Using the OID Monitor and Control Utilities To start the directory integration and provisioning server in non-SSL mode:

1. Be sure that OID Monitor is running. To verify this on UNIX, 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 "The OID Monitor (oidmon) Syntax" on page A-4.

2. Start the directory integration and provisioning server by using the OID Control Utility. Do this by entering:

   ```
   oidctl [connect=connect_string] server=odisrv [instance=instance_number]
   [config=configuration_set_number] [flags="[host=hostname] [port=port_number]
   [debug=debug_level] [refresh=interval_between_refresh]
   [grpID=group_identifier_of_provisioning_profile]
   [maxprofiles=number_of_profiles]
   [ sslauth=ssl_mode ]"] start
   ```

Table A–5 describes the arguments in this command.

Table A–5 Description of Arguments for Starting the Oracle Directory Integration and Provisioning Server

Argument	Description
`connect=`*`connect_string`*	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 and provisioning 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 directory integration and provisioning 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 and provisioning server **See Also:** Table 10–2 on page 10-7 for a description of the various debug levels
`refresh=`*`interval_between_refreshes`*	Specifies the interval, in minutes, between server refreshes 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

Table A–5 (Cont.) Description of Arguments for Starting the Oracle Directory Integration and Provisioning Server

Argument	Description

`sslauth=ssl_mode`	SSL modes:
	■ 0: SSL is not used—that is, non-SSL mode
	■ 1: SSL used for encryption only—that is, with no PKI authentication. A wallet is not used in this case.
	■ 2: SSL is used with one-way authentication. This mode requires you to specify a complete path name of an Oracle Wallet, including the file name itself, unlike other Oracle Internet Directory tools that expect only the wallet location. For example, in a server-only installation, or in a complete installation, you would enter something like this:
	`oidctl server=odisrv [instance=instance_number] [configset=configset_number] [grpID=group_identifier_of_provisioning_profile] flags="host=myhost port=myport sslauth=2`
	In a client-only installation, you would enter something like this:
	`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] [refresh=interval_between_refresh] [maxprofiles=number_of_profiles] [sslauth=ssl_mode]`

Starting the Oracle Directory Integration and Provisioning Server Without Using the OID Monitor and the OID Control Utility In a client-only installation, where the OID Monitor and OID Control tools are not available, the Oracle directory integration and provisioning server can 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 A–5 on page A-13 remain the parameters for each type of invocation.

To start the directory integration and provisioning server, enter the following at the command line:

```
odisrv [host=host_name] [port=port_number]
config=configuration_set_number [instance=instance_number] [debug=debug_level]
[refresh=interval_between_refresh] [maxprofiles=number_of_profiles]
[sslauth=ssl_mode]
```

Stopping the Oracle Directory Integration and Provisioning Server

The way you stop the directory integration and provisioning server depends on the tool that you used to start it.

Stopping the Oracle Directory Integration and Provisioning Server by Using OID Monitor and the OID Control Utility If you started the directory integration and provisioning 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 and provisioning 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 "The OID Monitor (oidmon) Syntax" on page A-4.

2. Stop the directory integration and provisioning server by entering:

   ```
   oidctl [connect=connect_string] server=odisrv instance=instance stop
   ```

Stopping the Oracle Directory Integration and Provisioning Server Without Using OID Monitor and the OID Control Utility In a client-only installation, where the OID Monitor and OID Control tools are not available, the Oracle directory integration and provisioning server can be started without OID Control. To stop the server without these tools, use the `stopodiserver.sh` tool, which is located in the *$ORACLE_HOME*`/ldap/admin` directory.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.3.2.2-1 or later. Visit:
  `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit:
  `http://www.datafocus.com/`

See Also: "The StopOdiServer.sh Tool Syntax" on page A-124 for instructions about using the stopodiserver.sh tool

Note: If the Oracle directory integration and provisioning server is stopped by any means other than the methods mentioned in this section, then 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 using the following command:

`$`*`ORACLE_HOME`*`/ldap/admin/stopodiserver.sh [-host `*`directory_server_host`*`] [-port `*`directory_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`**

Restarting Oracle Internet Directory Server Instances

When you want to refresh the server cache immediately, rather than at the next scheduled time, use the `RESTART` command. When the Oracle Internet Directory server restarts, it maintains the same parameters it had before it stopped.

To restart an Oracle Internet Directory server instance, at the system prompt, type:

```
oidctl connect=connect_string server={oidldapd|oidrepld|odisrv}
instance=server_instance_number  restart
```

OID Monitor must be running whenever you restart directory server instances.

If you try to contact a server that is not running, 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.

Starting and Stopping Oracle Internet Directory Servers on Either a Virtual Host or a Rack Node

When starting a directory server, a directory replication server, or a directory integration and provisioning server, use the `host` parameter to specify the virtual host name.

Starting and Stopping a Directory Server on Either a Virtual Host or a Rack Node

To start a directory server on a virtual host:

```
oidctl [connect=connect_string] host=virtual_host_name server=oidldapd
instance=instance_number configset=configset_number flags= "..." start
```

To stop a directory server on a virtual host:

```
oidctl host=virtual_host_name server=oidldapd instance=instance_number stop
```

Starting and Stopping a Directory Replication Server on Either a Virtual Host or a Rack Node

To start a directory replication server on a virtual host:

```
oidctl [connect=connect_string] host=virtual_host_name server=oidrepld
instance=instance_number flags= "..." start
```

To stop a directory replication server on a virtual host:

```
oidctl host=virtual_host_name server=oidrepld instance=instance_number stop
```

Starting and Stopping a Oracle Directory Integration and Provisioning Server on Either a Virtual Host or a Rack Node

To start a directory integration and provisioning server on a virtual host:

```
oidctl [connect=connect_string] host=virtual_host_name server=odisrv
instance=instance_number configset=configset_number flags= "..." start
```

To stop a directory integration and provisioning server on a virtual host:

```
oidctl host=virtual/host_name server=odisrv instance=instance_number stop
```

When the directory server is started to run on the virtual host, it binds and listens to requests on the specified LDAP port on the IP address or IP addresses that correspond to the virtual host only.

When communicating with the directory server, the directory replication server uses the virtual host name. Further, the `replicaID` attribute that represents the unique replication identification for the Oracle Internet Directory node is generated once. It is independent of the host name and hence requires no special treatment in cold failover configuration.

When communicating with the directory server, the directory integration and provisioning server uses the virtual host name.

Entry and Attribute Management Command-Line Tools Syntax

This section tells you how to use the following tools:

- The Catalog Management Tool (catalog.sh) Syntax
- ldapadd Syntax
- ldapaddmt Syntax

- ldapbind Syntax
- ldapcompare Syntax
- ldapdelete Syntax
- ldapmoddn Syntax
- ldapmodify Syntax
- ldapmodifymt Syntax
- ldapsearch Syntax

Note: Various UNIX shells interpret some characters—for example, asterisks (*)—as special characters. Depending on the shell you are using, you may need to escape these characters.

The Catalog Management Tool (catalog.sh) 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

If you want to use additional attributes in search filters, then 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.

Before running catalog.sh, be sure that the directory server is either stopped or in read-only mode. Otherwise, data will be inconsistent.

Caution: Do not use the catalog.sh `-delete` option on indexes created by the Oracle Internet Directory base schema. Removing indexes from base schema attributes can adversely impact the operation of Oracle Internet Directory.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

The Catalog Management tool uses this syntax:

`catalog.sh -connect` *`connect_string`* `{-add|-delete} {-attr` *`attr_name`*`|-file` *`file_name`*`}`

Table A–6 Arguments for the Catalog Management Tool (catalog.sh)

Argument	Description
`-connect` *`connect_string`*	Specifies the connect string to connect to the directory database. This argument is mandatory. **See Also:** *Oracle9i Net Services Administrator's Guide* in the Oracle Database Documentation Library
`-add -attr` *`attr_name`*	Indexes the specified attribute
`-delete -attr` *`attr_name`*	Drops the index from the specified attribute
`-add -file` *`file_name`*	Indexes attributes (one for each line) in the specified file
`-delete -file` *`file_name`*	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:

- "The OID Control Utility (oidctl) Syntax" on page A-6 and for instructions on starting and restarting directory servers. Note that OID Monitor must be running before you start a directory server.
- "The OID Monitor (oidmon) Syntax" on page A-4 for information about starting OID Monitor
- "Matching Rules" on page B-47 for the matching rules supported by Oracle Internet Directory

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-31.

See Also: "Adding Configuration Set Entries by Using ldapadd" on page 5-7 for an explanation of using ldapadd to configure a server with an input file

ldapadd uses this syntax:

```
ldapadd [arguments] -f file_name
```

where *file_name* 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
```

Table A–7 Arguments for ldapadd

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.)

Table A–7 (Cont.) Arguments for ldapadd

Optional Arguments	Description
`-D "binddn"`	When authenticating to the directory, specifies doing so as the entry specified in *binddn*—that is, the DN of the user seeking authentication. Use this with the `-w` *password* option.
`-E "character_set"`	Specifies native character set encoding. See Chapter G, "Globalization Support in the Directory".
`-f file_name`	Specifies the input name of the LDIF format import data file. For a detailed explanation of how to format an LDIF file, see "LDAP Data Interchange Format (LDIF) Syntax" on page A-2.
`-h ldaphost`	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
`-K`	Same as `-k`, but performs only the first step of the Kerberos bind
`-k`	Authenticates using Kerberos authentication instead of simple authentication. To enable this option, you must compile with KERBEROS defined.You must already have a valid ticket granting ticket.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
`-n`	Shows what would occur without actually performing the operation
`-O ref_hop_limit`	Specifies the number of referral hops that a client should process. The default value is 5.
`-p directory_server_port_number`	Connects to the directory on TCP port *directory_server_port_number*. If you do not specify this option, the tool connects to the default port (389).
`-P wallet_password`	Specifies wallet password required for one-way or two-way SSL connections
`-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

Table A–7 (Cont.) Arguments for ldapadd

Optional Arguments	Description
`-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 UNIX, 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"`
`-X dsml_file`	Specifies the input name of the DSML format import data file.

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 file_name
```

where *file_name* 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.

Table A–8 Arguments for ldapaddmt

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*—that is, the DN of the user seeking authentication. Use this with the `-w password` option.
`-E "character_set"`	Specifies native character set encoding. See Chapter G, "Globalization Support in the Directory"
`-h ldap_host`	Connects to *ldaphost*, rather than to the default host, that is, your local computer. *ldaphost* can be a computer name or an IP address.
`-K`	Same as -k, but performs only the first step of the kerberos bind
`-k`	Authenticates using Kerberos authentication instead of simple authentication. To enable this option, you must compile with KERBEROS defined. You must already have a valid ticket granting ticket.
`-M`	Instructs the tool to send the `ManageDSAIT` control to the server. The `ManageDSAIT` control instructs the server not to send referrals to clients. Instead a referral entry is returned as a regular entry.
`-n`	Shows what would occur without actually performing the operation.
`-O ref_hop_limit`	Specifies the number of referral hops that a client should process. The default value is 5.
`-p ldapport`	Connects to the directory on TCP port *ldapport*. If you do not specify this option, the tool connects to the default port (389).
`-P wallet_password`	Specifies wallet password required for one-way or two-way SSL connections
`-T`	Sets the number of threads for concurrently processing entries

Table A–8 (Cont.) Arguments for ldapaddmt

Optional Arguments	Description
`-U SSLAuth`	Specifies SSL Authentication Mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
`-v`	Specifies verbose mode
`-V ldap_version`	Specifies the version of the LDAP protocol to use. The default value is 3, which causes the tool to use the LDAP v3 protocol. A value of 2 causes the tool to use the LDAP v2 protocol.
`-w password`	Provides the password required to connect
`-W wallet_location`	Specifies wallet location required for one-way or two-way SSL connections. For example, on UNIX, 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"
`-X dsml_file`	Specifies the input name of the DSML format import data file.

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]
```

Table A–9 Arguments for ldapbind

Arguments	Description
-D "*binddn*"	When authenticating to the directory, specifies doing so as the entry specified in *binddn*—that is, the DN of the user seeking authentication. Use this with the `-w` *password* option.
-E "*.character_set*"	Specifies native character set encoding. See Chapter G, "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.

Table A–9 (Cont.) Arguments for ldapbind

Arguments	Description
-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 UNIX, 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”
-O *sasl_security_properties*	Specifies SASL security properties. The security property supported is -O "auth". This security property is for DIGEST-MD5 SASL mechanism. It enables authentication with no data integrity or data privacy.
-Y *sasl_mechanism*	Specifies a SASL mechanism. These mechanisms are supported: ■ Y "DIGEST-MD5" ■ Y "EXTERNAL": The SASL authentication in this mechanism is done on top of two-way SSL authentication. In this case the identity of the user stored in the SSL wallet is used for SASL authentication.
-R *sasl_realm*	Specifies a SASL realm

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
```

Table A–10 Arguments for ldapcompare

Optional Arguments	Description
-a *attribute name*	Specifies the attribute on which to perform the compare. This argument is mandatory.
-b *"basedn"*	Specifies the distinguished name of the entry on which to perform the compare. This argument is mandatory.
-v *attribute value*	Specifies the attribute value to compare. This argument is mandatory.
-D *binddn*	When authenticating to the directory, specifies doing so as the entry is specified in *binddn*—that is, the DN of the user seeking authentication. 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 10-6.
-E *"character_set"*	Specifies native character set encoding. See Chapter G, "Globalization Support in the Directory".
-f *file_name*	Specifies the input file name
-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

Table A–10 Arguments for ldapcompare

Optional Arguments	Description
-U *SSLAuth*	Specifies SSL authentication mode: ■ 1 for no authentication required ■ 2 for one way authentication required ■ 3 for two way authentication required
-V *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 UNIX, 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_file_name]
```

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"
```

Table A–11 Arguments for ldapdelete

Optional Argument	Description
-D *"binddn"*	When authenticating to the directory, uses a full DN for the *binddn* parameter—that is, the DN of the user seeking authentication; 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 10-6.

Table A–11 (Cont.) Arguments for ldapdelete

Optional Argument	Description
-E "*character_set*"	Specifies native character set encoding. See Chapter G, "Globalization Support in the Directory".
-f *input_file_name*	Specifies the input file name
-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.
`-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 UNIX, 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"
```

Table A–12 Arguments for ldapmoddn

Argument	Description
-b *"basedn"*	Specifies DN of the entry to be moved. This argument is mandatory.
-D *"binddn"*	When authenticating to the directory, do so as the entry is specified in *binddn*—that is, the DN of the user seeking authentication. Use this with the `-w` *password* option.
-E *"character_set"*	Specifies native character set encoding. See Chapter G, "Globalization Support in the Directory".
-f *file_name*	Specifies the input file name
-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. Either this argument or the -R argument must be specified.
-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

Table A–12 Arguments for ldapmoddn

Argument	Description
-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. Either this argument or the -N argument must be specified.
-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 UNIX, 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 file_name
```

where *file_name* 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. These arguments are all optional.

Table A–13 Arguments for ldapmodify

Argument	Description
-a	Denotes that entries are to be added, and that the input file is in LDIF format.

Table A–13 (Cont.) Arguments for ldapmodify

Argument	Description
-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*—that is, the DN of the user seeking authentication. Use this with the `-w` *password* option.
-E "*character_set*"	Specifies native character set encoding. See Chapter G, "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).
-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

Table A–13 (Cont.) Arguments for ldapmodify

Argument	Description
-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 UNIX, 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 in this section:

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 previously in Line 1. 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
  ```

To specify an attribute as single-valued, include in the attribute definition entry in the LDIF file the keyword SINGLE-VALUE with surrounding white space.

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 file_name
```

where *file_name* 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-31 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.

The arguments in the following table are all optional.

Table A–14 Arguments for ldapmodifymt

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.)
`-D "`*`binddn`*`"`	When authenticating to the directory, specifies doing so as the entry is specified in *binddn*—that is, the DN of the user seeking authentication. Use this with the `-w` *password* option.
`-E "`*`character_set`*`"`	Specifies native character set encoding. See Chapter G, "Globalization Support in the Directory".

Table A–14 (Cont.) Arguments for ldapmodifymt

Argument	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.
`-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.
`-W wallet_location`	Specifies wallet location required for one-way or two-way SSL connections. For example, on UNIX, 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.

The ldapsearch tool uses this syntax:

```
ldapsearch [arguments] filter [attributes]
```

The *filter* format must be compliant with RFC-2254.

> **See Also:** RFC-2254 available at `http://www.ietf.org` 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.
- Various UNIX shells interpret some characters—for example, asterisks (*)—as special characters. Depending on the shell you are using, you may need to escape these characters.

Table A–15 Arguments for ldapsearch

Argument	Description
-b *"basedn"*	Specifies the base DN for the search. This argument is mandatory.
-s *scope*	This argument is mandatory. Specifies search scope: base, one, or sub 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
-A	Retrieves attribute names only (no values)
-a *deref*	Specifies alias dereferencing: never, always, search, or find
-B	Allows printing of non-ASCII values

Table A–15 (Cont.) Arguments for ldapsearch

Argument	Description
-D "*binddn*"	When authenticating to the directory, specifies doing so as the entry specified in *binddn*—that is, the DN of the user seeking authentication. Use this with the `-w` *password* option.
-d *debug level*	Sets debugging level to the level specified (see Table 10–2 on page 10-7)
-E "*character_set*"	Specifies native character set encoding. See Chapter G, "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

Table A–15 (Cont.) Arguments for ldapsearch

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 UNIX, 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
-X	Prints the entries in DSML v1 format.

Examples of ldapsearch Filters

Study the following examples to see how to build your own search commands.

Example 1: Base Object Search The following example performs a base-level search on the directory from the root.

```
ldapsearch -p 389 -h myhost -b "" -s base -v "objectclass=*"
```

- `-b` specifies base DN for the search, root in this case.
- `-s` specifies whether the search is a base search (`base`), one level search (`one`) or subtree search (`sub`).
- `"objectclass=*"` specifies the filter for search.

Example 2: One-Level Search The following example performs a one level search starting at `"ou=HR, ou=Americas, o=IMC, c=US"`.

```
ldapsearch -p 389 -h myhost -b "ou=HR, ou=Americas, o=IMC, c=US" -s one -v
"objectclass=*"
```

Example 3: Subtree Search The following example performs a subtree search and returns all entries having a DN starting with `"cn=us"`.

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub -v "cn=Person*"
```

Example 4: Search Using Size Limit The following example actually retrieves only two entries, even if there are more than two matches.

```
ldapsearch -h myhost -p 389 -z 2 -b "ou=Benefits,ou=HR,ou=Americas,o=IMC,c=US"
-s one "objectclass=*"
```

Example 5: Search with Required Attributes The following example returns only the `DN` attribute values of the matching entries:

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub -v "objectclass=*" dn
```

The following example retrieves only the distinguished name along with the surname (`sn`) and description (`description`) attribute values:

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub -v "cn=Person*" dn sn description
```

Example 6: Search for Entries with Attribute Options The following example retrieves entries with common name (`cn`) attributes that have an option specifying a language code attribute option. This particular example retrieves entries in which the common names are in French and begin with the letter R.

```
ldapsearch -p 389 -h myhost -b "c=US" -s sub "cn;lang-fr=R*"
```

Suppose that, in the entry for John, no value is set for the `cn;lang-it` language code attribute option. In this case, the following example does not return John's entry:

```
ldapsearch -p 389 -h myhost -b "c=us" -s sub "cn;lang-it=Giovanni"
```

Example 7: Searching for All User Attributes and Specified Operational Attributes The following example retrieves all user attributes and the `createtimestamp` and `orclguid` operational attributes:

```
ldapsearch -p 389 -h myhost -b "ou=Benefits,ou=HR,ou=Americas,o=IMC,c=US" -s sub
"cn=Person*" * createtimestamp orclguid
```

The following example retrieves entries modified by Anne Smith:

```
ldapsearch -h sun1 -b "" "(&(objectclass=*)(modifiersname=cn=Anne
Smith))"
```

The following example retrieves entries modified between 01 April 2001 and 06 April 2001:

```
ldapsearch -h sun1 -b "" "(&(objectclass=*)(modifytimestamp >= 20000401000000)
(modifytimestamp <= 20000406235959))"
```

Note: Because `modifiersname` and `modifytimestamp` are not indexed attributes, use catalog.sh to index these two attributes. Then, restart the Oracle directory server before issuing the two previous ldapsearch commands.

Other Examples: Each of the following examples searches on port 389 of host sun1, and searches the whole subtree starting from the DN `"ou=hr,o=acme,c=us"`.

The following example searches for all entries with any value for the objectclass attribute.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree "objectclass=*"
```

The following example searches for all entries that have `orcl` at the beginning of the value for the `objectclass` attribute.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"objectclass=orcl*"
```

The following example searches for entries where the `objectclass` attribute begins with `orcl` and `cn` begins with foo.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"(&(objectclass=orcl*)(cn=foo*))"
```

The following example searches for entries in which the common name (`cn`) is not `foo`.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree "(!(cn=foo))"
```

The following example searches for entries in which `cn` begins with `foo` or `sn` begins with `bar`.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"(|(cn=foo*)(sn=bar*))"
```

The following example searches for entries in which `employeenumber` is less than or equal to 10000.

```
ldapsearch -p 389 -h sun1 -b "ou=hr, o=acme, c=us" -s subtree
"employeenumber<=10000"
```

Bulk Operations Command-Line 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 1.3.2.2-1 or later. Visit:
  `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit:
  `http://www.datafocus.com/`

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.

Note: Make sure that when bulkmodify is invoked, server-side entry cache is disabled.

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 connect_string -base "base_dn" -size number_of_entries
```

`-encode "character_set"`

Table A–16 Arguments for bulkdelete

Mandatory Argument	Description
`-connect connect_string`	Specifies the connect string to connect to the directory database. This argument is mandatory. **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. This argument is mandatory.
`-size number_of_entries`	Specifies the number of entries to be committed as a part of one transaction.
`-encode "character_set"`	Specifies native character set encoding. **See Also:** Chapter G, "Globalization Support in the Directory".

bulkload Syntax

The bulkload command-line tool is useful for loading large number of entries to a directory server. It uses Oracle SQL*Loader to load directory entries. The bulkload tool expects the input file to be in LDIF.

See Also:

- "LDAP Data Interchange Format (LDIF) Syntax" on page A-2
- *Oracle Application Server 10g Upgrading to 10g (9.0.4)* for any special instructions about upgrading orclguids when bulkloading an LDIF file from an older version of Oracle Internet Directory

The bulkload tool performs its operation in following phases:

1. Check

   In the check phase, all entries of LDIF files are verified for valid LDAP schema and duplicate entries. If there are any errors reported by bulkloader, then the user needs to rectify the error and retry bulkload.

2. Generate

In the generate phase, the LDIF input is converted into intermediate files that can be used by SQL*Loader to load the data into the Oracle Internet Directory directory store.

3. Load

   The Intermediate files generated in generate phase are loaded into the Oracle9*i* Database Server which is the Oracle Internet Directory directory store. Bulkloader supports two types of loading of data:

   - Incremental Mode Loading

     In incremental mode, you can append data to existing directory data. It should be used when you want to append a small amount of data. It is faster than other "add" methods, but slower than bulk mode loading. In this mode, Bulkloader does not drop and rebuild catalog indexes. Instead, it uses SQL*Loader in insert mode to add data to the database and update indexes through inserts.

     Here, "small amount" is a relative number, it really depends upon existing data in directory as well amount of data to be loaded and the hardware capabilities to handle the load.

     To invoke incremental mode, user need to specify -append along with other options.

     It is important to note that when using bulkload in incremental mode, one must put the directory server in read-modify mode. During read-modify mode, search and modify operations are allowed but add, delete, and modifyDN operations are restricted.

     See Also: "Task 2: Configure Structural Access Items" on page 14-21 for instructions on using Oracle Directory Manager to set access rights

   - Bulk Mode Loading

     In bulk mode, you must be able to add or append large number of entries to a directory. By default, Bulkloader runs in bulk mode. Bulk mode is faster than incremental mode.

     In bulk mode, all Oracle Internet Directory server instances should be stopped. In this mode, Bulkloader drops existing indexes and recreates them after loading of data. For data loading, it uses SQL*Loader direct-path mode.

See Also: "Stopping an Oracle Directory Server Instance" on page A-9

4. Index Creation

   After the load is complete, the indexes are recreated if the load was done in “bulk” mode. Also, the Bulkloader tool provides an option just to recreate all indexes. This is useful in case if previous index creation was unsuccessful for some reason.

5. Directory Data Recovery

   A failure in the 'load' phase of bulkload operation can leave directory data in inconsistent state. Bulkloader can revert back to original state that existed prior to the invocation of bulkload. Use the -recover option to recover directory data in case of Bulkload failure.

Usage Scenarios for the bulkload Tool

The bulkload tool can be used in single node as well as multiple node environments.

Single Node Environment

Loading in 'bulk' mode The typical usage scenario is to load directory data after Oracle Internet Directory installation. One would want to 'check' the LDIF file for schema errors, 'generate' the intermediate files and 'load' the data into the Oracle Internet Directory directory store. The 'parallel' option is normally faster since the load and index creation happens in parallel. The invocation of bulkload will be something like:

```
bulkload.sh -connect <conn_str> -check -generate -load -parallel <LDIF>
```

One can break up the above operation one into separate 'check, 'generate' and 'load' invocations. The 'check' can also be avoided if the LDIF data is from another Oracle Internet Directory directory node.

Loading in 'incremental' or 'append' Mode If one needs to add directory entries to an Oracle Internet Directory directory store that already has some user LDIF data, then the 'incremental' or 'append' mode is the way to go. This mode is normally faster than other methods of adding entries to the directory. However, it must be ensured that the Oracle Internet Directory LDAP instances are in read-modify mode

before bulkload begins to append data. The typical invocation of bulkload will be something like:

```
bulkload.sh -connect <conn_str> -check -generate -load -append <LDIF>
```

Index recreation The bulkload operation will take care of either updating the indexes or creating the indexes. However, due to issues like improper sizing, it may so happen that the indexes may not be updated or created properly. Bulkloader provides an option to recreate all the indexes. The invocation of bulkload will be:

```
bulkload.sh -connect <conn_str> -index
```

Data recovery on errors Due to issues like improper disk sizing, the 'load' phase of bulkload may fail. If this happens, there are chances that the directory data is inconsistent and hence, bulkloader provides an option to recover the directory data to the state that existed prior to the invocation of bulkload:

```
bulkload.sh -connect <conn_str> -recover
```

Multi-Node Environment

Bulk Mode Loading One must specify the connect strings of all the Oracle Internet Directory nodes involved. The bulkload will be invoked something like:

```
bulkload.sh -connect "<conn_str1> <conn_str2> <conn_str3>" -check -generate
-load -parallel <LDIF>
```

The bulkloader will handle this as given below:

1. Bulkloader prompts the user to put all Oracle Internet Directory LDAP servers on all the nodes in 'read-modify' mode.
2. Bulkloader will prompt the user to bring down the Oracle Internet Directory servers on the node corresponding to <conn_str1>
3. The 'check' and 'generate' will be performed on the node corresponding to <cons_str1>
4. The 'load' will be performed on the node corresponding to <conns_str1>
5. Bulkloader will prompt the user to bring up the Oracle Internet Directory servers on the node corresponding to <conn_str1>
6. Steps 2, 4 and 5 will be repeated for all the nodes.
7. Now, all the Oracle Internet Directory LDAP servers can be changed to read-write mode.

Incremental Mode Loading Here, the approach is the same as in the bulk mode loading except that the Oracle Internet Directory LDAP servers need not be shutdown, rather all of them need to be in 'read-only' mode.

Limitations of Bulkloader in Oracle Internet Directory 10*g* (9.0.4)

In multi-node environments, it is the directory administrator's responsibility to make sure that all nodes have same schema before running bulkloader.

If the user sees bad entries logged in badentry.ldif but does not rectify the entries, then data can be inconsistent.

The 'check' mode of bulkloader does not check and report the lack of parent/child relationships between entries.

The 'incremental' or 'append' mode is only for adding new entries not new attribute values to existing entries.

In multi-node environments, the first connect string specified must refer to the local node.

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

Syntax for the bulkload Tool

The bulkload tool uses this syntax:

```
bulkload.sh -connect connect_string <[-check] [-generate] [-restore]
[-numThread] [-parallel] [-encode] [-append] [-load] | [-index] | [-recover]
absolute_path_to_LDIF_data_file
```

Table A–17 lists and describes the arguments.

Table A–17 Arguments for bulkload.sh

Argument	Description	Mandatory?
`-connect`	Specifies the net service name defined in the `tnsnames.ora` file. For loading data in single node, specify its connect-string—for example `orcl`. For loading data in multiple nodes, specify connect-strings of all nodes—for example, `orcl1 orcl2 orcl3` **See Also:** *Oracle9i Net Services Administrator's Guide* in the Oracle9*i* Database Server Documentation Library	Yes
`-check`	Checks LDAP schema for inconsistencies and for existence of duplicate DNs in the file	No
`-generate`	Creates intermediate files suitable for loading into Oracle Internet Directory using SQL*Loader	No
`-restore`	Assumes operational attributes, such as `orclguid`, `creatorsname`, and `createtimestamp`, are already present in the specified LDIF file. When used with `-generate`, bulkload.sh avoids creating duplicate operational attribute values in the output SQL*Loader files. When used with `-check`, bulkload.sh suppresses errors associated with finding pre-existing operational attribute values in LDIF files.	No
`-numThread n`	Specifies the number of threads to be created. - numThread is useful only in -generate mode. The default value is number of CPUs on machine + 1	No
`-parallel`	Specifies that the loading should be done in parallel. Useful with -load option	No
`-encode`	Specifies native character set encoding **See Also:** Chapter G, "Globalization Support in the Directory"	No
`-append`	Specifies append incremental mode (Default is bulkmode append)	No
`-load`	Loads files resulting from generate phase into specified database	No
`-index`	Recreates indexes on all catalog tables	No
`-recover`	In case of bulkload.sh failure, recovers directory with original data	No
`-file_name`	Absolute path of ldif file	No

The LDIF data file path must be fully specified for check or generate operations.

While calling bulkload at least one of `-check`, `-generate`, `-load`, `-recover` or `-index` actions must be specified.

There are certain combinations of options that should be called together for effective bulkloading.

- The -restore flag should only be used when ldif file contains operational attributes such as `orclguid`, `creatorsname`, and so forth.
- The path name to the LDIF data file should be fully specified, and the data file must be specified for the -check or `-generate` actions.
- `-numThread` is useful only if given with -generate option.
- `-parallel` should be called with `-load` only.
- `-recover` or `-index` should not be specified with any other option.

> **See Also:** "Directory Replication" for further information and resources for bulk loading multiple nodes in a replicated environment

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 Administration" on page 25-1

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.

Make sure that when bulkmodify is invoked, server side entry cache is disabled.

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 connect_string -b "base_dn" {-a|-r} attr_name -v att_value [-f
filter] [-s size]
```

Table A–18 Arguments for bulkmodify

Argument	Description
`-c connect_string`	Specifies the connect string for the directory database. This argument is mandatory. **See Also:** *Oracle9i Net Services Administrator's Guide* in the Oracle Database Documentation Library

Table A–18 (Cont.) Arguments for bulkmodify

Argument	Description
`-b "base_dn"`	Specifies the base DN of the subtree to be modified. This argument is mandatory.
`-a attr_name`	Specifies the attribute name for addition. This argument is mandatory.
`-r attr_name`	Specifies the attribute name for replacement. This argument is mandatory.
`-v attr_value`	Specifies the attribute value for either addition or replacement. This argument is mandatory.
`-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 G, "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 connect_string -b "base_DN" -f file_name
```

Table A–19 Arguments for ldifwrite

Mandatory Argument	Description
`-c connect_string`	Specifies the net service name for the directory that is the source of the data, as defined in the `tnsnames.ora` file. This argument is mandatory. **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. This argument is mandatory. If the base DN is the replication agreement entry, then you can back up part of the naming context based on the LDAP naming context configuration. In this case, the syntax is: `ldifwrite -c connect string -b "replication agreement DN" -f file name` **See Also:** "Rules for Partial Replication Filtering" on page 24-35
`-f file_name`	Specifies the name of the LDIF file to be created. This argument is mandatory.
-E "`character_set`"	Specifies native character set encoding. **See Also:** "Using Globalization Support with ldifwrite" on page G-9

Example 1: Converting All Entries Under a Specified Naming Context to an LDIF File

This 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 Oracle Internet Directory password, enter the password of the underlying ODS user. The default password is `ods`.

Example 2: Converting Part of a Specified Naming Context to an LDIF File

This example uses the following naming context objects defined in partial replication:

```
dn: cn=includednamingcontext000001,
cn=replication namecontext,
orclagreementid=000001,
orclreplicaid=node replica identifier,
cn=replication configuration
orclincludednamingcontexts: c=us
orclexcludednamingcontexts: ou=Americas, c=us
orclexcludedattributes: userpassword
objectclass: top
objectclass: orclreplnamectxconfig
```

In this example, all entries under `c=us` are backed up except `ou=Americas,c=us`. The `userpassword` attribute is also excluded. The command is

```
ldifwrite -c connect string -b "cn=includednamingcontext000001,cn=replication
namecontext,orclagreementid=000001,orclreplicaid=node replica
identifier,cn=replication configuration" -f file name
```

Replication-Management Command-Line Tools Syntax

This section contains these topics:

- Replication Conflict Resolution Command-Line Tools
- The Replication Environment Management Tool

Replication Conflict Resolution Command-Line 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 queue
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.

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 change log 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 1.3.2.2-1 or later. Visit:
  `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit:
  `http://www.datafocus.com/`

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 connect_string [-start change_number]
[-end change_number] [-equal change_number] -supplier supplier_node
```

The arguments are:

Table A–20 Arguments for Moving a Change from the Human Intervention Queue into the Retry Queue

Argument	Description
`-connect` *connect_string*	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 connect_string [-start change_number] [-end change_number]
[-equal change_number] -supplier supplier_node
```

Arguments are:

Table A–21 Arguments for Moving a Change from the Human Intervention Queue into the Purge Queue

Argument	Description
`-connect` *connect_string*	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.

Table A–21 (Cont.) Arguments for Moving a Change from the Human Intervention Queue into the Purge Queue

Argument	Description
`-end` *change_number*	Specifies the end change number for the purge operation. If you skip this option, then the command moves all the changes with change numbers greater or equal to the specified start change number back to the purge queue.
`-equal` *change_number*	Specifies the change number of the change. The command moves the exact change conflict back to the purge queue. This option should not be present when `-start` or `-end` is used.
`-supplier` *supplier_node*	Specifies the supplier node where the changes originate

Note: When using hiqretry.sh or hiqpurge.sh, if you do not want all changes to be moved, then you must supply either the `-equal` flag, or a combination of the `-start` and `-end` flags.

Examples: Using the Human Intervention Queue Manipulation Tool The following examples illustrate how to use the Human Intervention Queue Manipulation Tool.

Example: Retrying and Discarding Changes Suppose that, after analyzing the replication log, you decide to do the following:

- Retry changes coming from the supplier node, ldap_rep1, with change numbers between 10324 to 10579
- Discard changes with change numbers between 10581 to 10623.

To do this, you issue these two commands:

```
hiqretry.sh -connect oiddb1 -start 10324 -end 10579 -supplier ldap_rep1
hiqpurge.sh -connect oiddb1 -start 10581 -end 10623 -supplier ldap_repl
```

The first command moves changes originating in ldap_rep1 with change numbers from 10324 to 10579 back to the retry queue. The second command deletes changes that originate in the supplier ldap_repl and that have change numbers from 10581 to 10623.

Example: Moving a Single Change from the Human Intervention Queue to the Retry Queue The following command moves the change with change number equal to 10519 back to the retry queue.

```
hiqretry.sh -connect oiddb1 -equal 10519 -supplier ldap_repl
```

Example: Moving a Group of Changes from the Human Intervention Queue to the Retry Queue The following command moves all the changes with change number greater or equal to 10324 back to the retry queue.

```
hiqretry.sh -connect oiddb1 -start 10324 -supplier ldap_repl
```

The following command moves all the changes with change numbers less than or equal to 10579 back to the retry queue.

```
hiqretry.sh -connect oiddb1 -end 10579 -supplier ldap_repl
```

Example: Moving All Changes from the Human Intervention Queue to the Retry Queue The following command includes no options. It moves all changes that originate in the supplier ldap_repl from the human intervention queue to the retry queue.

```
hiqretry.sh -connect oiddb1 -supplier ldap_repl
```

The OID Reconciliation Tool

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 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

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]
```

Table A–22 Arguments for Reconciling Inconsistent Data by Using the OID Reconciliation Tool

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. **Note:** You cannot specify `base` or `one-level` for this argument.
`-b "basedn"`	Specifies the distinguished name of the entry on which to perform reconciliation.
`-W supplier_password`	The password of the replication DN of the supplier node
`-w consumer_password`	The password of the replication DN of the consumer node
`-T thread`	Number of worker threads

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
```

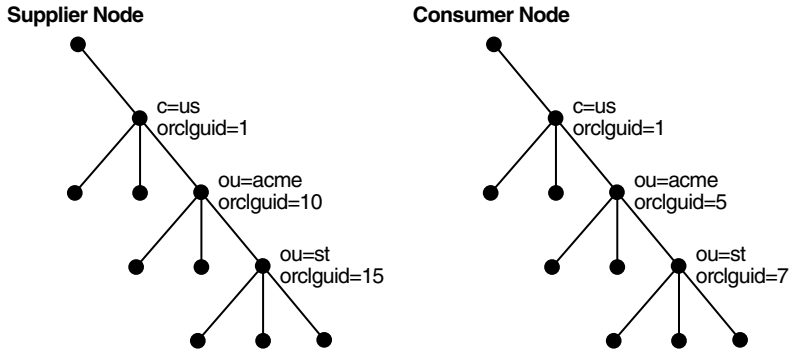
If the global identification (orclGuid) of both parents ("o=acme,c=us") match, and -s subtree is not set, then the OID Reconciliation Tool replaces only the entry itself on the consumer node with the specified entry from the supplier node.

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 that 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 providing 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 the `basedn` argument for oidreconcile.

The Replication Environment Management Tool

The Replication Environment Management Tool is used to manage Oracle Internet Directory replication configuration activities.

More specifically, the replication environment management tool:

- Configures Oracle9*i* Advanced Replication-based multimaster replication
- Scans the replication environment and verify the correctness of replication setup of the Oracle9*i* Advanced Replication-based DRG
- Rectifies any problem in the Oracle9*i* Advanced Replication-based DRG. If the tool cannot rectify a problem, it reports the point or points of failure to you for manual intervention
- Reports queue statistics, deferred transactions errors, and administrative request errors of a Oracle9*i* Advanced Replication-based DRG
- Reconfigures the Oracle9*i* Advanced Replication-based DRG
- Configures LDAP-based replication
- Reconfigures the LDAP-based DRG

The syntax for the Replication Environment Management Tool is:

```
remtool [ -asrsetup | -addnode | -delnode |-asrverify  |-asrrectify | -chgpwd |
-asrcleanup | -suspendasr | -resumeasr  | -dispqstat  |  -dispasrerr | -paddnode
| -pdelnode  | -pchgpwd  | -presetpwd  | -pchgwalpwd | -pcleanup  ]
[-v] [-connect repadmin_name/password@net_service_name  |
-bind host:port/replication_dn_password]
```

Table A–23 Arguments for the Replication Environment Management Tool (remtool)

Argument	Description
`-connect`	For Oracle9*i* Advanced Replication only. Connect string of the master definition site (MDS) or Remote Master Site (RMS) only. If `-connect` option is not specified, then the tool prompts you for connection details. This argument requires three elements: ■ Name of the replication administrator ■ Password of the replication administrator ■ Net service name of the MDS or RMS
`-bind`	For LDAP based replication only. Bind details of the directory server This argument requires three elements ■ Host name at which directory server is running ■ Port at which directory server is listening ■ Password of replication DN.
`-v`	Verbose mode Specifying `-v` option not only shows the progress of remtool, but also logs all actions of remtool in `remtool.log` created under `$ORACLEHOME/ldap/log` folder. If `-v` option is not specified, then remtool logs only limited action of remtool.

Table A–24 Options for Configuring and Managing an Oracle9i Advanced Replication-Based DRG (remtool)

Argument	Description
`-asrsetup`	Create a Directory Replication Group (DRG) by configuring Oracle9*i* Advanced Replication
`-addnode`	Add a new node to an existing DRG.
`-delnode`	Reconfigure Oracle9*i* Advanced Replication to delete a node from an existing DRG
`-asrverify`	Verify correctness of Oracle9*i* Advanced Replication configuration of a DRG. This option reports problems but does not rectify them.
`-asrrectify`	Verify correctness of Oracle9*i* Advanced Replication setup for a DRG and rectify the problems, if any

Table A–24 (Cont.) Options for Configuring and Managing an Oracle9i Advanced Replication-Based DRG (remtool)

Argument	Description
`-chgpwd`	Change replication administrator database account password on all nodes of a DRG
`-asrcleanup`	Clean up Oracle9*i* Advanced Replication setup of a DRG
`-suspendasr`	Quiesce / Suspend replication activity of a DRG
`-resumeasr`	Resume replication activity of a DRG
`-dispqstat`	Display queue statistics of all nodes
-dispasrerr	Display all deferred transaction errors and administrative request errors of a DRG

Table A–25 Options for Configuring and Managing an LDAP-Based Replication DRG (remtool)

Argument	Description
`-paddnode`	Add a partial replica to a DRG.
`-pdelnode`	Delete a partial replica from a DRG
`-pchgpwd`	Change password of replication DN of a replica
`-presetpwd`	Reset password of replication DN of a replica
`-pchgwalpwd`	Change password of replication DN of a replica only in wallet
`-pcleanup`	Cleanup partial replication setup of a DRG

Example 1: Verifying Oracle9*i* Advanced Replication Configuration (Verbose Mode)

In the following example, the Replication Environment Management Tool:

- Verifies the correctness of Oracle9*i* Advanced Replication configuration of a DRG
- Reports on the verification as it progresses
- Does not rectify the problems

The command is:

```
remtool -asrverify -v
```

Example 2: Verifying Oracle9*i* Advanced Replication Configuration (Non-Verbose Mode)

In the following example, the Replication Environment Management Tool:

- Verifies the correctness of Oracle9*i* Advanced Replication configuration of a DRG
- Does not report on the verification as it progresses
- Does not rectify the problems

The command is:

```
remtool -asrverify
```

Example 3: Verifying Oracle9*i* Advanced Replication Configuration and Rectifying the Problems

In this example, the Replication Environment Management Tool:

- Verifies the correctness of Oracle9i Advanced Replication configuration of a DRG
- Reports on the verification as it progresses
- Rectifies the problems

The command is:

```
remtool -asrrectify -v -connect repadmin/repadmin@node_1.my_company.com
```

-ADDNODE Option

The syntax is:

```
remtool -addnode [-v] [-conn[ect] rep_admin_name/rep_admin_password@connectid_
of_mds_or_rms]
```

Usage Notes for the -ADDNODE Option

1. The `addnode` option is used to add a new node to an existing DRG created by `ASRSETUP` option.
2. The node to be added must be empty.
3. Oracle Internet Directory processes on the master definition site (MDS) and other remote master sites (RMSs) must be down.
4. After the addnode procedure is complete, Oracle Internet Directory processes can be started.

5. The SYSTEM user password of the new node is required for this option.

Example: -ADDNODE Option

In this example, MY_HOST3.MY_COMPANY.COM is added to a DRG consisting of MY_HOST1.MY_COMPANY.COM and MY_HOST2.MY_COMPANY.COM for which the following command is issued:

```
remtool -addnode -v -conn repadmin/repadmin@MY_HOST1.MY_COMPANY.COM
```

The results are:

```
MY_HOST1.MY_COMPANY.COM is Master Definition Site (MDS). Connected to MDS.
MY_HOST2.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
Directory Replication Group (DRG) details :

-------- ------------ ---------------------- ------------ ------------ ----
Instance Host Name     Global Name            Version      Replicaid    Site
Name                                                                    Type
-------- ------------ ---------------------- ------------ ------------ ----
rid2      my_host      MY_HOST1.MY_COMPANY.COM  OID 9.0.4.0.0 my_host_rid1  MDS
rid2      my_host      MY_HOST2.MY_COMPANY.COM  OID 9.0.4.0.0 my_host_rid2  RMS
-------- ------------ ---------------------- ------------ ------------ ----
Do you want to continue? [y/n] : y

-------------------------------------------------------------------------------
WARNING:
Make sure that the replication administrator database
account does not exist already in the new node to be
added to the DRG. If the account exists, that
account will be dropped and will be created newly.
-------------------------------------------------------------------------------
Enter global name of new node to be added          : MY_HOST3.MY_COMPANY.COM

Enter SYSTEM user password of new node to be added :
-------------------------------------------------------------------------------
Adding a new node...

MY_HOST3.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST3.MY_COMPANY.COM : Dropping replication administrator repadmin...
MY_HOST3.MY_COMPANY.COM : Creating replication administrator repadmin...
MY_HOST3.MY_COMPANY.COM : Granting privileges or roles required for replication
administrator to repadmin...
MY_HOST3.MY_COMPANY.COM : Granting privileges or roles required for replication
administrator to repadmin...
MY_HOST3.MY_COMPANY.COM : Granting privileges or roles required for replication
```

```
administrator to repadmin...
MY_HOST3.MY_COMPANY.COM : Dropping replication group LDAP_REP...
MY_HOST3.MY_COMPANY.COM : Creating purge job...
MY_HOST3.MY_COMPANY.COM : Dropping database link made to MY_HOST1.MY_
COMPANY.COM...
MY_HOST3.MY_COMPANY.COM : Dropping database link made to MY_HOST1.MY_
COMPANY.COM...
MY_HOST3.MY_COMPANY.COM : Creating database link to MY_HOST1.MY_COMPANY.COM...
MY_HOST3.MY_COMPANY.COM : Scheduling push job to MY_HOST1.MY_COMPANY.COM...
MY_HOST3.MY_COMPANY.COM : Dropping database link made to MY_HOST2.MY_
COMPANY.COM...
MY_HOST3.MY_COMPANY.COM : Dropping database link made to MY_HOST2.MY_
COMPANY.COM...
MY_HOST3.MY_COMPANY.COM : Creating database link to MY_HOST2.MY_COMPANY.COM...
MY_HOST3.MY_COMPANY.COM : Scheduling push job to MY_HOST2.MY_COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Dropping database link made to MY_HOST3.MY_
COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Dropping database link made to MY_HOST3.MY_
COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Creating database link to MY_HOST3.MY_COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Scheduling push job to MY_HOST3.MY_COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Dropping database link made to MY_HOST3.MY_
COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Dropping database link made to MY_HOST3.MY_
COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Creating database link to MY_HOST3.MY_COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Scheduling push job to MY_HOST3.MY_COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Quiescing replication activity...
MY_HOST1.MY_COMPANY.COM : Adding replication site MY_HOST3.MY_COMPANY.COM to
replication group LDAP_REP...
MY_HOST1.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST3.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST1.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST1.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST3.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST1.MY_COMPANY.COM : Resuming replication activity...
MY_HOST1.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST2.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST3.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST1.MY_COMPANY.COM : Verifying replication agreement entry...
MY_HOST1.MY_COMPANY.COM : Inserting replication agreement entry my_host_rid3...
CORRECTED:
MY_HOST1.MY_COMPANY.COM : "my_host_rid3" hostname has been added to replication
agreement entry.
MY_HOST2.MY_COMPANY.COM : Verifying replication agreement entry...
```

```
MY_HOST2.MY_COMPANY.COM : Inserting replication agreement entry my_host_rid3...
CORRECTED:
MY_HOST2.MY_COMPANY.COM : "my_host_rid3" hostname has been added to replication
agreement entry.
MY_HOST3.MY_COMPANY.COM : Verifying replication agreement entry...
MY_HOST3.MY_COMPANY.COM : Inserting replication agreement entry my_host_rid...
CORRECTED:
MY_HOST3.MY_COMPANY.COM : "my_host_rid" hostname has been added to replication
agreement entry.
MY_HOST3.MY_COMPANY.COM : Inserting replication agreement entry my_host_rid2...
CORRECTED:
MY_HOST3.MY_COMPANY.COM : "my_host_rid2" hostname has been added to replication
agreement entry.
MY_HOST3.MY_COMPANY.COM : Inserting replication agreement entry my_host_rid3...
CORRECTED:
MY_HOST3.MY_COMPANY.COM : "my_host_rid3" hostname has been added to replication
agreement entry.
MY_HOST1.MY_COMPANY.COM : Verifying initialization parameter...
MY_HOST2.MY_COMPANY.COM : Verifying initialization parameter...
MY_HOST3.MY_COMPANY.COM : Verifying initialization parameter...
--------------------------------------------------------------------------------
Node MY_HOST3.MY_COMPANY.COM has been added to this DRG.
--------------------------------------------------------------------------------
Directory Replication Group (DRG) details :

-------- ------------- ---------------------- ------------- ------------- ----
Instance Host Name     Global Name            Version       Replicaid     Site
Name                                                                      Type
-------- ------------- ---------------------- ------------- ------------- ----
rid1     my_host       MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
rid2     my_host       MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2  RMS
rid3     my_host       MY_HOST3.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid3  RMS
-------- ------------- ---------------------- ------------- ------------- ----
```

-ASRSETUP Option

The syntax is:

```
remtool -asrsetup [-v]
```

Usage Notes for the -ASRSETUP Option

1. For the `-asrsetup` option, the `-conn[ect]` option is ignored.
2. The user is prompted for following details:

   ```
   MDS Globalname
   ```

```
MDS Password
globalname of all RMSs
password of all RMSs
```

3. All Oracle Internet Directory processes must be down in MDS and all RMSs. After the ASRSETUP option is completed, the user can bring up all Oracle Internet Directory processes and replication server processes.

Example: -ASRSETUP Option

In this example, a DRG is created consisting of MY_HOST1.MY_COMPANY.COM and MY_HOST2.MY_COMPANY.COM for which the following command is issued:

```
remtool -asrsetup -v
```

The results are as follows:

```
--------------------------------------------------------------------------------
ASR Setup for OID Replication
WARNING:
Make sure that the replication administrator that you
enter below does not exist already in any of the nodes
that will be part of the DRG to be created now. If the
user exists, that user will be dropped and will be
created newly.
--------------------------------------------------------------------------------
Enter replication administrator's name       : repadmin

Enter replication administrator's password   :
Reenter replication administrator's password :
Enter Master Definition Site (MDS) details   :
Enter global name of MDS                     : MY_HOST1.MY_COMPANY.COM

Enter SYSTEM user password of MDS            :
Enter Remote Master Site (RMS) details       :
Enter global name of RMS #  1                : MY_HOST2.MY_COMPANY.COM

Enter SYSTEM user password of MDS            :
Are there more Remote Master Sites in the group? [y/n/q] : n

Verify the details you had entered.
--------------------------------------------------------------------------------
Replication administrator's name   : repadmin
Master Definition Site             : MY_HOST1.MY_COMPANY.COM
Remote Master Site #  1            : MY_HOST2.MY_COMPANY.COM
Are these details correct? [y/n/q] : y
```

```
--------------------------------------------------------------------------------
ASR setup in progress...

MY_HOST1.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST1.MY_COMPANY.COM : Dropping replication administrator repadmin...
MY_HOST1.MY_COMPANY.COM : Creating replication administrator repadmin...
MY_HOST1.MY_COMPANY.COM : Granting privileges or roles required for replication
administrator to repadmin...
MY_HOST1.MY_COMPANY.COM : Granting privileges or roles required for replication
administrator to repadmin...
MY_HOST1.MY_COMPANY.COM : Granting privileges or roles required for replication
administrator to repadmin...
MY_HOST1.MY_COMPANY.COM : Creating purge job...
MY_HOST1.MY_COMPANY.COM : Dropping database link made to MY_HOST2.MY_
COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Dropping database link made to MY_HOST2.MY_
COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Creating database link to MY_HOST2.MY_COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Scheduling push job to MY_HOST2.MY_COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST2.MY_COMPANY.COM : Dropping replication administrator repadmin...
MY_HOST2.MY_COMPANY.COM : Creating replication administrator repadmin...
MY_HOST2.MY_COMPANY.COM : Granting privileges or roles required for replication
administrator to repadmin...
MY_HOST2.MY_COMPANY.COM : Granting privileges or roles required for replication
administrator to repadmin...
MY_HOST2.MY_COMPANY.COM : Granting privileges or roles required for replication
administrator to repadmin...
MY_HOST2.MY_COMPANY.COM : Creating purge job...
MY_HOST2.MY_COMPANY.COM : Dropping database link made to MY_HOST1.MY_
COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Dropping database link made to MY_HOST1.MY_
COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Creating database link to MY_HOST1.MY_COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Scheduling push job to MY_HOST1.MY_COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Dropping replication group LDAP_REP...
MY_HOST1.MY_COMPANY.COM : Creating replication group LDAP_REP...
MY_HOST1.MY_COMPANY.COM : Adding object TABLE ODS.ASR_CHG_LOG to replication
group LDAP_REP...
MY_HOST1.MY_COMPANY.COM : Generating replication support for TABLE ODS.ASR_CHG_
LOG...
MY_HOST1.MY_COMPANY.COM : Adding object TABLE ODS.ODS_CHG_STAT to replication
group LDAP_REP...
MY_HOST1.MY_COMPANY.COM : Generating replication support for TABLE ODS.ODS_CHG_
```

```
STAT...
MY_HOST2.MY_COMPANY.COM : Dropping replication group LDAP_REP...
MY_HOST1.MY_COMPANY.COM : Adding replication site MY_HOST2.MY_COMPANY.COM to
replication group LDAP_REP...
MY_HOST1.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST2.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST1.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST1.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST2.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST1.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST2.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST1.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST2.MY_COMPANY.COM : Executing deferred administrative requests...
MY_HOST1.MY_COMPANY.COM : Verifying initialization parameter...
MY_HOST2.MY_COMPANY.COM : Verifying initialization parameter...
MY_HOST1.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST2.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST1.MY_COMPANY.COM : Verifying replication agreement entry...
MY_HOST1.MY_COMPANY.COM : Inserting replication agreement entry my_host_...
CORRECTED:
MY_HOST1.MY_COMPANY.COM : "my_host_rid" hostname has been added to replication
agreement entry.
MY_HOST1.MY_COMPANY.COM : Inserting replication agreement entry my_host_rid2...
CORRECTED:
MY_HOST1.MY_COMPANY.COM : "my_host_rid2" hostname has been added to replication
agreement entry.
MY_HOST2.MY_COMPANY.COM : Verifying replication agreement entry...
MY_HOST2.MY_COMPANY.COM : Inserting replication agreement entry my_host_rid...
CORRECTED:
MY_HOST2.MY_COMPANY.COM : "my_host_rid1" hostname has been added to replication
agreement entry.
MY_HOST2.MY_COMPANY.COM : Inserting replication agreement entry my_host_rid2...
CORRECTED:
MY_HOST2.MY_COMPANY.COM : "my_host_rid2" hostname has been added to replication
agreement entry.
MY_HOST1.MY_COMPANY.COM : Resuming replication activity...
--------------------------------------------------------------------------------
ASR setup has been configured successfully.
--------------------------------------------------------------------------------
Directory Replication Group (DRG) details :

-------- ------------ ---------------------- ------------ ------------ ----
Instance Host Name     Global Name            Version      Replicaid    Site
Name                                                                    Type
-------- ------------ ---------------------- ------------ ------------ ----
```

```
rid1      my_host      MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
rid2      my_host      MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2  RMS
-------- ------------- ---------------------- ------------- ------------- ----
```

-CHGPWD Option

The syntax is:

```
remtool -chgpwd [-v] [-conn[ect] rep_admin_name/rep_admin_password@connectid_of_
mds_or_rms]
```

1. Used for changing password of replication administrator of DRG created by `ASRSETUP` procedure.

   ```
   created by ASRSETUP procedure.
   ```

2. In ASR based replication repadmin password is same in all nodes. This option will change the password of replication administrator database account at all nodes.

Example: -CHGPWD Option

In this example, the password of the replication administrator of a DRG consisting of `MY_HOST1.MY_COMPANY.COM` and `MY_HOST2.MY_COMPANY.COM` is changed for which, the following command is issued:

```
remtool -chgpwd -v -conn repadmin/repadmin@MY_HOST1.MY_COMPANY.COM
```

The results are:

```
MY_HOST1.MY_COMPANY.COM is Master Definition Site (MDS). Connected to MDS.
MY_HOST2.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
Directory Replication Group (DRG) details :

-------- ------------- ---------------------- ------------- ------------- ----
Instance Host Name     Global Name            Version       Replicaid     Site
Name                                                                      Type
-------- ------------- ---------------------- ------------- ------------- ----
rid1       my_host     MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
rid2       my_host     MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2  RMS
-------- ------------- ---------------------- ------------- ------------- ----
Enter new password of the replication administrator   :
Reenter new password of the replication administrator :
------------------------------------------------------------------------------
Changing the password of all nodes...

MY_HOST1.MY_COMPANY.COM : Changing password of replication administrator
```

```
repadmin...
MY_HOST2.MY_COMPANY.COM : Changing password of replication administrator
repadmin...
MY_HOST1.MY_COMPANY.COM : Dropping database link made to MY_HOST2.MY_
COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Dropping database link made to MY_HOST2.MY_
COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Creating database link to MY_HOST2.MY_COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Dropping database link made to MY_HOST1.MY_
COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Dropping database link made to MY_HOST1.MY_
COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Creating database link to MY_HOST1.MY_COMPANY.COM...
-------------------------------------------------------------------------------
Password has been changed.
-------------------------------------------------------------------------------
```

-DELNODE Option

The syntax is:

```
remtool -delnode [-v] [-conn[ect] rep_admin_name/rep_admin_password@connectid_
of_mds_or_rms]
```

Usage Notes for the -DELNODE Option

1. The -DELNODE option is used for deleting a node from a DRG that is created by ASRSETUP option.
2. The global name of the node to be deleted must be specified.
3. Oracle Internet Directory processes must be down in all nodes of the DRG.
4. The -DELNODE option can be used to remove only RMS from a DRG.
5. The -DELNODE option cannot be used to remove MDS from a DRG.
6. The -DELNODE option can be also used to remove a RMS from a DRG that has only two nodes: one MDS and one RMS. This leaves the DRG with only an MDS. The user can add multimaster nodes later on to this DRG.
7. If remtool detects that one of the nodes in a DRG is not up and running when invoked with the -DELNODE option, remtool selects that node for deletion.

Example 1: -DELNODE Option

In this example, MY_HOST3.MY_COMPANY.COM is removed from a DRG consisting of MY_HOST1.MY_COMPANY.COM, MY_HOST2.MY_COMPANY.COM and MY_HOST3.MY_COMPANY.COM for which the following command is issued:

```
remtool -delnode -v -conn repadmin/repadmin@MY_HOST1.MY_COMPANY.COM
```

The results are:

```
MY_HOST1.MY_COMPANY.COM is Master Definition Site (MDS). Connected to MDS.
MY_HOST2.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
MY_HOST3.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
Directory Replication Group (DRG) details :

-------- ------------- ---------------------- ------------ ------------- ----
Instance Host Name     Global Name            Version      Replicaid     Site
Name                                                                     Type
-------- ------------- ---------------------- ------------ ------------- ----
rid1      my_host      MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
rid2      my_host      MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2  RMS
rid3      my_host      MY_HOST3.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid3  RMS
-------- ------------- ---------------------- ------------ ------------- ----
Do you want to continue? [y/n] : y

Enter globalname of node to be deleted : MY_HOST3.MY_COMPANY.COM

------------------------------------------------------------------------------
Deleting an existing node...

MY_HOST1.MY_COMPANY.COM : Dropping replication site MY_HOST3.MY_COMPANY.COM from
replication group LDAP_REP...
MY_HOST3.MY_COMPANY.COM : Dropping replication group LDAP_REP...
MY_HOST3.MY_COMPANY.COM : Unscheduling push job to MY_HOST1.MY_COMPANY.COM...
MY_HOST3.MY_COMPANY.COM : Unscheduling push job to MY_HOST2.MY_COMPANY.COM...
MY_HOST3.MY_COMPANY.COM : Dropping database link made to MY_HOST1.MY_
COMPANY.COM...
MY_HOST3.MY_COMPANY.COM : Dropping database link made to MY_HOST2.MY_
COMPANY.COM...
MY_HOST3.MY_COMPANY.COM : Dropping database link made to MY_HOST1.MY_
COMPANY.COM...
MY_HOST3.MY_COMPANY.COM : Dropping database link made to MY_HOST2.MY_
COMPANY.COM...
Enter "SYSTEM" user password for "MY_HOST3.MY_COMPANY.COM" database at "my_host"
host :
MY_HOST3.MY_COMPANY.COM : Dropping replication administrator repadmin...
MY_HOST1.MY_COMPANY.COM : Unscheduling push job to MY_HOST3.MY_COMPANY.COM...
```

```
MY_HOST1.MY_COMPANY.COM : Dropping database link made to MY_HOST3.MY_
COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Dropping database link made to MY_HOST3.MY_
COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Unscheduling push job to MY_HOST3.MY_COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Dropping database link made to MY_HOST3.MY_
COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Dropping database link made to MY_HOST3.MY_
COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST2.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST1.MY_COMPANY.COM : Verifying replication agreement entry...
MY_HOST1.MY_COMPANY.COM : Deleting replication agreement entry my_host_rid3...
CORRECTED:
MY_HOST1.MY_COMPANY.COM : "my_host_rid3" hostname has been removed from
replication agreement entry as it is not part of DRG or was repeated.
MY_HOST2.MY_COMPANY.COM : Verifying replication agreement entry...
MY_HOST2.MY_COMPANY.COM : Deleting replication agreement entry my_host_rid3...
CORRECTED:
MY_HOST2.MY_COMPANY.COM : "my_host_rid3" hostname has been removed from
replication agreement entry as it is not part of DRG or was repeated.
-------------------------------------------------------------------------------
Node MY_HOST3.MY_COMPANY.COM has been deleted from this DRG.
-------------------------------------------------------------------------------
Directory Replication Group (DRG) details :

-------- ------------ ---------------------- ------------ ------------ ----
Instance Host Name     Global Name            Version      Replicaid    Site
Name                                                                    Type
-------- ------------ ---------------------- ------------ ------------ ----
rid1     my_host       MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
rid2     my_host       MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2  RMS
-------- ------------ ---------------------- ------------ ------------ ----

===============================================================================
```

Example 2: -DELNODE Option

In this example, `MY_HOST2.MY_COMPANY.COM` is removed from a DRG consisting of `MY_HOST1.MY_COMPANY.COM` and `MY_HOST2.MY_COMPANY.COM` for which the following command is issued:

```
remtool -delnode -v -conn repadmin/repadmin@MY_HOST1.MY_COMPANY.COM
```

The results are:

```
MY_HOST1.MY_COMPANY.COM is Master Definition Site (MDS). Connected to MDS.
MY_HOST2.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
Directory Replication Group (DRG) details :

-------- ------------ ---------------------- ------------ ------------ ----
Instance Host Name     Global Name            Version      Replicaid    Site
Name                                                                    Type
-------- ------------ ---------------------- ------------ ------------ ----
rid1     my_host      MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
rid2     my_host      MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2  RMS
-------- ------------ ---------------------- ------------ -----------------
Do you want to continue? [y/n] : y

Enter globalname of node to be deleted : MY_HOST2.MY_COMPANY.COM

------------------------------------------------------------------------------
Deleting an existing node...

MY_HOST1.MY_COMPANY.COM : Dropping replication site MY_HOST2.MY_COMPANY.COM from
replication group LDAP_REP...
MY_HOST2.MY_COMPANY.COM : Dropping replication group LDAP_REP...
MY_HOST2.MY_COMPANY.COM : Unscheduling push job to MY_HOST1.MY_COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Dropping database link made to MY_HOST1.MY_
COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Dropping database link made to MY_HOST1.MY_
COMPANY.COM...
Enter "SYSTEM" user password for "MY_HOST2.MY_COMPANY.COM" database at "my_host"
host :
MY_HOST2.MY_COMPANY.COM : Dropping replication administrator repadmin...
MY_HOST1.MY_COMPANY.COM : Unscheduling push job to MY_HOST2.MY_COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Dropping database link made to MY_HOST2.MY_
COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Dropping database link made to MY_HOST2.MY_
COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST1.MY_COMPANY.COM : Verifying replication agreement entry...
MY_HOST1.MY_COMPANY.COM : Deleting replication agreement entry my_host_rid2...
CORRECTED:
MY_HOST1.MY_COMPANY.COM : "my_host_rid2" hostname has been removed from
replication agreement entry as it is not part of DRG or was repeated.
------------------------------------------------------------------------------
Node MY_HOST2.MY_COMPANY.COM has been deleted from this DRG.
------------------------------------------------------------------------------
Directory Replication Group (DRG) details :
```

```
-------- ------------ ---------------------- ------------ ------------ ----
Instance Host Name     Global Name            Version      Replicaid    Site
Name                                                                    Type
-------- ------------ ---------------------- ------------ ------------ ----
rid1      my_host       MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
-------- ------------ ---------------------- ------------ ------------ ----
Warning : This replication group has only one node.
```

-ASRCLEANUP Option

The syntax is:

```
remtool -asrcleanup [-v] [-conn[ect] rep_admin_name/rep_admin_
password@connectid_of_mds_or_rms]
```

Usage Notes for the -ASRCLEANUP Option

1. The -ASRCLEANUP option is used to clean up an existing ASR setup.
2. The -ASRCLEANUP option can be used to clean up flawed ASR setup as well.
3. The -ASRCLEANUP option prompts the user for SYSTEM password of all sites taking part in replication.

Example 1: -ASRCLEANUP Option

In this example, ASR setup is cleaned up from a DRG consisting of MY_HOST1.MY_COMPANY.COM and MY_HOST2.MY_COMPANY.COM for which the following command is issued:

```
remtool -asrcleanup -v
```

The results are:

```
Enter replication administrator's name       : repadmin

Enter replication administrator's password   :
Enter global name of MDS                     : my_host1.my_company.com

MY_HOST1.MY_COMPANY.COM is Master Definition Site (MDS). Connected to MDS.
MY_HOST2.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
Directory Replication Group (DRG) details :

-------- ------------ ---------------------- ------------ ------------ ----
Instance Host Name     Global Name            Version      Replicaid    Site
Name                                                                    Type
```

```
-------- ------------ ---------------------- ------------ ------------ ----
rid1      my_host       MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
rid2      my_host       MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2  RMS
-------- ------------ ---------------------- ------------ ------------ ----
Do you want to continue? [y/n] : y

-------------------------------------------------------------------------------
Cleaning up...

MY_HOST1.MY_COMPANY.COM : Dropping replication site MY_HOST2.MY_COMPANY.COM from
replication group LDAP_REP...
MY_HOST2.MY_COMPANY.COM : Dropping replication group LDAP_REP...
MY_HOST2.MY_COMPANY.COM : Unscheduling push job to MY_HOST1.MY_COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Dropping database link made to MY_HOST1.MY_
COMPANY.COM...
MY_HOST2.MY_COMPANY.COM : Dropping database link made to MY_HOST1.MY_
COMPANY.COM...
Enter "SYSTEM" user password for "MY_HOST2.MY_COMPANY.COM" database at "my_host"
host :
MY_HOST2.MY_COMPANY.COM : Dropping replication administrator repadmin...
MY_HOST1.MY_COMPANY.COM : Dropping replication group LDAP_REP...
MY_HOST1.MY_COMPANY.COM : Unscheduling push job to MY_HOST2.MY_COMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Dropping database link made to MY_
HOST2.MYCOMPANY.COM...
MY_HOST1.MY_COMPANY.COM : Dropping database link made to MY_HOST2.MY_
COMPANY.COM...
Enter "SYSTEM" user password for "MY_HOST1.MY_COMPANY.COM" database at "my_host"
host :
MY_HOST1.MY_COMPANY.COM : Dropping replication administrator repadmin...
-------------------------------------------------------------------------------
ASR setup has been cleaned up.
-------------------------------------------------------------------------------
```

-ASRRECTIFY Option

The syntax is:

```
remtool -asrrectify [-v] [-conn[ect] rep_admin_name/rep_admin_
password@connectid_of_mds_or_rms]
```

Usage Notes for the -ASRRECTIFY Option

1. The -ASRRECTIFY option is used for detecting and rectifying problems in Oracle9*i* Advanced Replication setup.
2. The -ASRRECTIFY option reports errors and rectifies them.

3. Oracle Corporation recommends that, before executing this option, you stop Oracle Internet Directory servers.
4. To use the -ASRRECTIFY option, all the nodes must be up and running. The -ASRRECTIFY option fails, if any of the nodes are not running.
5. If necessary, the -ASRRECTIFY option prompts for the SYSTEM user password.

Example 1: -ASRRECTIFY Option

In this example, ASR setup errors are deducted and rectified in a DRG consisting of MY_HOST1.MY_COMPANY.COM and MY_HOST2.MY_COMPANY.COM for which the following command is issued:

```
remtool -asrrectify -v -conn repadmin/repadmin@my_host1.my_company.com

MY_HOST1.MY_COMPANY.COM is Master Definition Site (MDS). Connected to MDS.
MY_HOST2.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
Directory Replication Group (DRG) details :

-------- ------------ ---------------------- ------------ ------------ ----
Instance Host Name     Global Name            Version      Replicaid    Site
Name                                                                    Type
-------- ------------ ---------------------- ------------ ------------ ----
rid1      my_host      MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
rid2      my_host      MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2  RMS
-------- ------------ ---------------------- ------------ ------------ ----
Do you want to continue? [y/n] : y

--------------------------------------------------------------------------------
Rectifying ASR setup...

MY_HOST1.MY_COMPANY.COM : Verifying initialization parameter...
MY_HOST2.MY_COMPANY.COM : Verifying initialization parameter...
MY_HOST1.MY_COMPANY.COM : Verifying replication administrator roles...
MY_HOST2.MY_COMPANY.COM : Verifying replication administrator roles...
MY_HOST1.MY_COMPANY.COM : Verifying database links...
MY_HOST2.MY_COMPANY.COM : Verifying database links...
MY_HOST1.MY_COMPANY.COM : Verifying purge job...
MY_HOST2.MY_COMPANY.COM : Verifying purge job...
MY_HOST1.MY_COMPANY.COM : Verifying scheduled links...
MY_HOST2.MY_COMPANY.COM : Verifying scheduled links...
MY_HOST1.MY_COMPANY.COM : Verifying availability of replication objects...
MY_HOST2.MY_COMPANY.COM : Verifying availability of replication objects...
MY_HOST1.MY_COMPANY.COM : Verifying replication group...
MY_HOST1.MY_COMPANY.COM : Quiescing replication activity...
```

```
MY_HOST1.MY_COMPANY.COM : Adding object TABLE ODS.ASR_CHG_LOG to replication
group LDAP_REP...
MY_HOST1.MY_COMPANY.COM : Generating replication support for TABLE ODS.ASR_CHG_
LOG...
CORRECTED:
MY_HOST1.MY_COMPANY.COM : Replication support has been generated for TABLE
ODS.ASR_CHG_LOG.
MY_HOST1.MY_COMPANY.COM : Quiescing replication activity...
MY_HOST1.MY_COMPANY.COM : Adding object TABLE ODS.ODS_CHG_STAT to replication
group LDAP_REP...
MY_HOST1.MY_COMPANY.COM : Generating replication support for TABLE ODS.ODS_CHG_
STAT...
CORRECTED:
MY_HOST1.MY_COMPANY.COM : Replication support has been generated for TABLE
ODS.ODS_CHG_STAT.
MY_HOST1.MY_COMPANY.COM : Resuming replication activity...
MY_HOST2.MY_COMPANY.COM : Verifying replication group...
MY_HOST1.MY_COMPANY.COM : Resuming replication activity...
MY_HOST1.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST2.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST1.MY_COMPANY.COM : Verifying replication agreement entry...
MY_HOST2.MY_COMPANY.COM : Verifying replication agreement entry...

--------------------  -----  -----  -----  -----  -----  -----  -----
DB Name               Init   Repl   DB     Purge  Sch.   Repl   Repl
                      Param  Admin  Links  Job    Links  Group  Agrmt
                             Role                              Entry
--------------------  -----  -----  -----  -----  -----  -----  -----
MY_HOST1.MY_COMPANY.  Chkd   Chkd   Chkd   Chkd   Chkd   Crtd   Chkd
MY_HOST2.MY_COMPANY.  Chkd   Chkd   Chkd   Chkd   Chkd   Chkd   Chkd
--------------------  -----  -----  -----  -----  -----  -----  -----
Legends :
  Chkd  - Checked. No errors.
  Crtd  - ASR setup errors were found and corrected.
  Err   - Error occurred while doing ASR setup verification.
  NCrtd - ASR setup has errors, but not corrected.
-------------------------------------------------------------------------------
Summary of findings:
CORRECTED:
MY_HOST1.MY_COMPANY.COM : Replication support has been generated for TABLE
ODS.ASR_CHG_LOG.

CORRECTED:
MY_HOST1.MY_COMPANY.COM : Replication support has been generated for TABLE
ODS.ODS_CHG_STAT.
```

```
--------------------------------------------------------------------------------
```

Example 2: -ASRRECTIFY Option

In this example, ASR setup errors are deducted and rectified in a DRG consisting of MY_HOST1.MY_COMPANY.COM and MY_HOST2.MY_COMPANY.COM. Here remtool detects that user has changed global name of MY_HOST2.MY_COMPANY.COM to NEWNAME.MY_COMPANY.COM after setting up ASR. Remtool rectifies this error first before continuing with other checks. The following command is issued:

```
remtool -asrrectify -v -conn repadmin/repadmin@my_host1.my_company.com
```

The results are:

```
MY_HOST1.MY_COMPANY.COM is Master Definition Site (MDS). Connected to MDS.
Enter "SYSTEM" user password for "MY_HOST2.MY_COMPANY.COM" database at "my_host"
host :
NEWNAME.MY_COMPANY.COM : Renaming global name to MY_HOST2.MY_COMPANY.COM
(instance name : rid2, hostname : my_host)
CORRECTED:
MY_HOST2.MY_COMPANY.COM : Global name of database "rid2" at host "my_host" has
been changed to MY_HOST2.MY_COMPANY.COM.
MY_HOST2.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
CORRECTED:
MY_HOST2.MY_COMPANY.COM : Global name of database "rid2" at host "my_host" has
been changed to MY_HOST2.MY_COMPANY.COM.
Directory Replication Group (DRG) details :

-------- ------------ ---------------------- ------------ ------------ ----
Instance Host Name     Global Name            Version      Replicaid    Site
Name                                                                    Type
-------- ------------ ---------------------- ------------ ------------ ----
rid1      my_host      MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
rid2      my_host      MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2  RMS
-------- ------------ ---------------------- ------------ ------------ ----
Do you want to continue? [y/n] : y

--------------------------------------------------------------------------------
Rectifying ASR setup...

MY_HOST1.MY_COMPANY.COM : Verifying initialization parameter...
MY_HOST2.MY_COMPANY.COM : Verifying initialization parameter...
MY_HOST1.MY_COMPANY.COM : Verifying replication administrator roles...
MY_HOST2.MY_COMPANY.COM : Verifying replication administrator roles...
MY_HOST1.MY_COMPANY.COM : Verifying database links...
MY_HOST2.MY_COMPANY.COM : Verifying database links...
```

```
MY_HOST1.MY_COMPANY.COM : Verifying purge job...
MY_HOST2.MY_COMPANY.COM : Verifying purge job...
MY_HOST1.MY_COMPANY.COM : Verifying scheduled links...
MY_HOST2.MY_COMPANY.COM : Verifying scheduled links...
MY_HOST1.MY_COMPANY.COM : Verifying availability of replication objects...
MY_HOST2.MY_COMPANY.COM : Verifying availability of replication objects...
MY_HOST1.MY_COMPANY.COM : Verifying replication group...
MY_HOST1.MY_COMPANY.COM : Resuming replication activity...
MY_HOST2.MY_COMPANY.COM : Verifying replication group...
MY_HOST1.MY_COMPANY.COM : Resuming replication activity...
MY_HOST1.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST2.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST1.MY_COMPANY.COM : Verifying replication agreement entry...
MY_HOST2.MY_COMPANY.COM : Verifying replication agreement entry...

--------------------  -----  -----  -----  -----  -----  -----  -----
DB Name               Init   Repl   DB     Purge  Sch.   Repl   Repl
                      Param  Admin  Links  Job    Links  Group  Agrmt
                             Role                              Entry
--------------------  -----  -----  -----  -----  -----  -----  -----
MY_HOST1.MY_COMPANY.  Chkd   Chkd   Chkd   Chkd   Chkd   Chkd   Chkd
MY_HOST2.MY_COMPANY.  Chkd   Chkd   Chkd   Chkd   Chkd   Chkd   Chkd
--------------------  -----  -----  -----  -----  -----  -----  -----
Legends :
  Chkd  - Checked. No errors.
  Crtd  - ASR setup errors were found and corrected.
  Err   - Error occurred while doing ASR setup verification.
  NCrtd - ASR setup has errors, but not corrected.
-------------------------------------------------------------------------------
```

-ASRVERIFY Option

The syntax is:

```
remtool -asrverify [-v] [-conn[ect] rep_admin_name/rep_admin_password@connectid_
of_mds_or_rms]
```

Usage Notes for the -ASRVERIFY Option

1. This option is used for just detecting problems in ASR setup. It will just report errors and won't rectify them.
2. While executing this option, Oracle Internet Directory servers can be up.
3. If, by mistake, the replication administrator account is dropped in any of the nodes, then the `-asrverify`" option fails. In this case, the `-asrrectify`

option can be used to recreate the replication administrator account and add it back to the DRG.

4. If, by mistake, the password of replication administrator account of one node of the DRG to be checked is changed, then the `-asrverify` option fails. In this case, the `-asrrectify` option can be used to change the replication administrator account and add it back to the DRG.
5. If the global name of any node is changed after Oracle9*i* Advanced Replication setup, then the `-asrverify` reports an error and does not proceed further. The `-asrrectify` option can be used to revert back to the previous global name and rectify other issues.
6. To exercise this option, all the nodes must be up and running.

Example 1: -ASRVERIFY Option

In this example, errors in ASR setup are found in a DRG consisting of two nodes for which the following command is issued:

```
remtool -asrverify -v -conn repadmin/repadmin@my_host1.my_company.com
```

The results are:

```
MY_HOST1.MY_COMPANY.COM is Master Definition Site (MDS). Connected to MDS.
MY_HOST2.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
Directory Replication Group (DRG) details :

-------------------- ---------------------- ------------- ------------- ----
Instance Host Name     Global Name             Version       Replicaid     Site
Name                                                                       Type
-------------------- ---------------------- ------------- ------------- ----
rid1     my_host      MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1   MDS
rid2     my_host      MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2   RMS
------------------------------------------- ------------- -----------------
--------------------------------------------------------------------------------
Verifying ASR setup...

MY_HOST1.MY_COMPANY.COM : Verifying initialization parameter...
MY_HOST2.MY_COMPANY.COM : Verifying initialization parameter...
MY_HOST1.MY_COMPANY.COM : Verifying replication administrator roles...
MY_HOST2.MY_COMPANY.COM : Verifying replication administrator roles...
MY_HOST1.MY_COMPANY.COM : Verifying database links...
MY_HOST2.MY_COMPANY.COM : Verifying database links...
MY_HOST1.MY_COMPANY.COM : Verifying purge job...
MY_HOST2.MY_COMPANY.COM : Verifying purge job...
```

```
MY_HOST1.MY_COMPANY.COM : Verifying scheduled links...
MY_HOST2.MY_COMPANY.COM : Verifying scheduled links...
MY_HOST1.MY_COMPANY.COM : Verifying availability of replication objects...
MY_HOST2.MY_COMPANY.COM : Verifying availability of replication objects...
MY_HOST1.MY_COMPANY.COM : Verifying replication group...
ASR SETUP ERROR/WARNING:
MY_HOST1.MY_COMPANY.COM : Replication support is not available for TABLE
ODS.ASR_CHG_LOG.
ASR SETUP ERROR/WARNING:
MY_HOST1.MY_COMPANY.COM : Replication support is not available for TABLE
ODS.ODS_CHG_STAT.
MY_HOST2.MY_COMPANY.COM : Verifying replication group...
ASR SETUP ERROR/WARNING:
MY_HOST2.MY_COMPANY.COM : Replication support is not available for TABLE
ODS.ASR_CHG_LOG.
ASR SETUP ERROR/WARNING:
MY_HOST2.MY_COMPANY.COM : Replication support is not available for TABLE
ODS.ODS_CHG_STAT.
MY_HOST1.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST2.MY_COMPANY.COM : Verifying uniqueness of replication agreement entry...
MY_HOST1.MY_COMPANY.COM : Verifying replication agreement entry...
MY_HOST2.MY_COMPANY.COM : Verifying replication agreement entry...

--------------------  -----  -----  -----  -----  -----  -----  -----
DB Name               Init   Repl   DB     Purge  Sch.   Repl   Repl
                      Param  Admin  Links  Job    Links  Group  Agrmt
                             Role                              Entry
--------------------  -----  -----  -----  -----  -----  -----  -----
MY_HOST1.MY_COMPANY.  Chkd   Chkd   Chkd   Chkd   Chkd   NCrtd  Chkd
MY_HOST2.MY_COMPANY.  Chkd   Chkd   Chkd   Chkd   Chkd   NCrtd  Chkd
--------------------  -----  -----  -----  -----  -----  -----  -----
Legends :
  Chkd  - Checked. No errors.
  Crtd  - ASR setup errors were found and corrected.
  Err   - Error occurred while doing ASR setup verification.
  NCrtd - ASR setup has errors, but not corrected.
--------------------------------------------------------------------------------
Summary of findings:
ASR SETUP ERROR/WARNING:
MY_HOST1.MY_COMPANY.COM : Replication support is not available for TABLE
ODS.ASR_CHG_LOG.

ASR SETUP ERROR/WARNING:
MY_HOST1.MY_COMPANY.COM : Replication support is not available for TABLE
ODS.ODS_CHG_STAT.
```

```
ASR SETUP ERROR/WARNING:
MY_HOST2.MY_COMPANY.COM : Replication support is not available for TABLE
ODS.ASR_CHG_LOG.

ASR SETUP ERROR/WARNING:
MY_HOST2.MY_COMPANY.COM : Replication support is not available for TABLE
ODS.ODS_CHG_STAT.
--------------------------------------------------------------------------------
```

-DISPASRERR Option

The syntax is:

```
remtool -dispasrerr [-v] [-conn[ect] rep_admin_name/rep_admin_
password@connectid_of_mds_or_rms]
```

Usage Notes for the -DISPASRERR Option

1. This option is used for displaying ASR errors in a DRG.
2. It displays both ASR administrative request errors and deferred transaction errors.

Example: -DISPASRERR Option

In this example, ASR errors of DRG consisting of `MY_HOST1.MY_COMPANY.COM` and `MY_HOST2.MY_COMPANY.COM` are reported for which the following command is issued:

```
remtool -dispasrerr -v -conn repadmin/repadmin@my_host1.my_company.com

MY_HOST1.MY_COMPANY.COM is Master Definition Site (MDS). Connected to MDS.
MY_HOST2.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
Directory Replication Group (DRG) details :

-------- ------------ ---------------------- ------------ ------------ ----
Instance Host Name     Global Name            Version      Replicaid    Site
Name                                                                    Type
-------- ------------ ---------------------- ------------ ------------ ----
rid      my_host      MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1 MDS
rid2     my_host      MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2 RMS
-------- ------------ ---------------------- ------------ ------------ ----
--------------------------------------------------------------------------------
Following administrative request errors were found at MY_HOST1.MY_COMPANY.COM

-------------------  -------------------  ------------------------------
```

```
Admin request         Request raised at     Error
raised by
--------------------  --------------------  ------------------------------
REPADMIN              MY_HOST1.MY_COMPANY.  ORA-23309: object ODS.ASR_CHG_L
REPADMIN              MY_HOST1.MY_COMPANY.  ORA-23309: object ODS.ODS_CHG_S
REPADMIN              MY_HOST1.MY_COMPANY.  ORA-23416: table "ODS"."ODS_CHG
REPADMIN              MY_HOST1.MY_COMPANY.  ORA-23308: object ODS.ODS_CHG_S
REPADMIN              MY_HOST1.MY_COMPANY.  ORA-23416: table "ODS"."ASR_CHG
REPADMIN              MY_HOST1.MY_COMPANY.  ORA-23308: object ODS.ASR_CHG_L
--------------------  --------------------  ------------------------------
--------------------------------------------------------------------------------
Following deferred transaction errors were found at MY_HOST1.MY_COMPANY.COM

---------------  ---------------  ---------------  ----------------------------
Deferred         Deferred Trans   Destination      Error
Transaction ID   Origin DB
---------------  ---------------  ---------------  ----------------------------
1.2.3733         MY_HOST1.MY_COM  MY_HOST1.MY_COM  ORA-01403: no data found

---------------  ---------------  ---------------  ----------------------------
No deferred transaction errors were found at MY_HOST2.MY_COMPANY.COM
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
```

-DISPQSTAT Option

The syntax is:

```
remtool -dispqstat [-v] [-conn[ect] rep_admin_name/rep_admin_password@connectid_
of_mds_or_rms]
```

1. This option is used for displaying queue statistics of DRG that uses ASR based replication. This option cannot be used for the DRG that uses LDAP based replication.
2. For DRG that uses ASR and LDAP based replication, this option displays queue statistics for nodes that uses ASR based replication only.

Example: -DISPQSTAT Option

In this example, queue statistics of DRG consisting of `MY_HOST1.MY_COMPANY.COM` and `MY_HOST2.MY_COMPANY.COM` are reported for which the following command is issued:

```
remtool -dispqstat -v -conn repadmin/repadmin@my_host1.my_company.com
```

The results are:

```
MY_HOST1.MY_COMPANY.COM is Master Definition Site (MDS). Connected to MDS.
MY_HOST2.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
Directory Replication Group (DRG) details :

-------- ------------ ---------------------- ------------ ------------ ----
Instance Host Name     Global Name            Version      Replicaid    Site
Name                                                                    Type
-------- ------------ ---------------------- ------------ ------------ ----
rid1     my_host      MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
rid2     my_host      MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2  RMS
-------- ------------ ---------------------- ------------ ------------ ----
Queue Statistics :
------------- ------------- --------- --------- --------- --------- ---------
   Supplier       Consumer       New      Retry     Purge      HIQ    Change #
------------- ------------- --------- --------- --------- --------- ---------
MY_HOST1.MY CO MY_HOST1.MY CO     3         9         10         6      2003
MY_HOST1.MY CO MY HOST2.MY CO     2         7         8          5      2001
MY_HOST2.MY CO MY_HOST1.MY CO     2         8         5          8      2002
MY_HOST2.MY CO MY_HOST2.MY CO     2        10         7          8      2000
------------- ------------- --------- --------- --------- --------- ---------
Legends
  New: No. of new change logs
  Retry: No. of change logs in retry queue
  Purge: No. of change logs in purge queue
  HIQ: No. of change logs in Human Intervention Queue (HIQ)
  Change # : Last applied change log no.
```

-SUSPENDASR Option

The syntax is:

```
remtool -suspendasr [-v] [-conn[ect] rep_admin_name/rep_admin_
password@connectid_of_mds_or_rms]
```

Usage Notes for the -SUSPENDASR Option

1. This option is used to suspend Oracle9*i* Advanced Replication activity of a DRG that uses it for replication.
2. While Oracle9*i* Advanced Replication activity is suspended, replication cannot take place.

Example: -SUSPENDASR Option

In this example, replication activity of DRG consisting of `MY_HOST1.MY_COMPANY.COM` and `MY_HOST2.MY_COMPANY.COM` is suspended for which the following command is issued:

```
remtool -suspendasr -v -conn repadmin/repadmin@my_host1.my_company.com
```

The results are:

```
MY_HOST1.MY_COMPANY.COM is Master Definition Site (MDS). Connected to MDS.
MY_HOST2.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
Directory Replication Group (DRG) details :

-------- ------------- ---------------------- ------------ ------------- ----
Instance Host Name     Global Name            Version      Replicaid     Site
Name                                                                     Type
-------- ------------- ---------------------- ------------ ------------- ----
rid      my_host       MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
rid2     my_host       MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2  RMS
-------- ------------- ---------------------- ------------ ------------- ----
--------------------------------------------------------------------------------
Altering replication status...

MY_HOST1.MY_COMPANY.COM : Quiescing replication activity...
--------------------------------------------------------------------------------
Replication status has been altered successfully.
--------------------------------------------------------------------------------
```

-RESUMEASR Option

The syntax is:

```
remtool -resumeasr [-v] [-conn[ect] rep_admin_name/rep_admin_password@connectid_
of_mds_or_rms]
```

Usage Notes for the -RESUMEASR Option

1. This option is used to resume ASR activity of a DRG that uses ASR for replication.

Example: -RESUMEASR Option

In this example, replication activity of DRG consisting of `MY_HOST1.MY_COMPANY.COM` and `MY_HOST2.MY_COMPANY.COM` is resumed for which the following command is issued:

```
remtool -resumeasr -v -conn repadmin/repadmin@MY_HOST1.MY_COMPANY.COM
```

The results are:

```
MY_HOST1.MY_COMPANY.COM is Master Definition Site (MDS). Connected to MDS.
MY_HOST2.MY_COMPANY.COM is Remote Master Site (RMS). Connected to RMS.
Directory Replication Group (DRG) details :

-------- ------------ ---------------------- ------------ ------------ ----
Instance Host Name     Global Name            Version      Replicaid    Site
Name                                                                    Type
-------- ------------ ---------------------- ------------ ------------ ----
rid1      my_host      MY_HOST1.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid1  MDS
rid2      my_host      MY_HOST2.MY_COMPANY.COM OID 9.0.4.0.0 my_host_rid2  RMS
-------- ------------ ---------------------- ------------ ------------ ----
--------------------------------------------------------------------------------
Altering replication status...

MY_HOST1.MY_COMPANY.COM : Resuming replication activity...
--------------------------------------------------------------------------------
Replication status has been altered successfully.
--------------------------------------------------------------------------------
```

-PADDNODE Option

The syntax is:

```
remtool -paddnode [-v] [-bind <hostname>:<port>/<replication_dn_password>]
```

Usage Notes for the -PADDNODE Option

1. Using this option a read-only replica or a read-only partial replica can be added to a node, known as supplier node.
2. Supplier node can be part of a DRG that uses ASR for replication or LDAP for replication or both.
3. New replica to be added should not be part of any DRG.
4. If the user does not specify supplier directory details using -bind option, user is prompted to specify supplier details:
5. If the supplier details are valid, remtool identifies all nodes in the DRG, if any, and displays the details before asking for consumer details.
6. After getting consumer directory details, if the DRG has multiple nodes, it prompts the user to specify the supplier's replicaid. Here user can specify the replicaid of any node of the DRG that uses LDAP based replication.

7. In case user wants to specify a ASR based replica as supplier, user must specify the ASR based replica as supplier in -bind option or when remtool prompts the user to specify it.
8. Remtool, after adding a replica, displays a list of naming contexts available in supplier replica along with "*". "*" indicates that whole directory will be included for replication barring DSE. User can select to replicate a portion of directory by selecting required naming contexts or whole directory by selecting "*". If user does not select any naming context, none of the naming contexts will take part in replication.
9. Remtool includes `cn=oraclecontext` naming context for replication whether or not user specifies naming context(s) to be included for replication.

Example 1:-PADDNODE Option

In this example, the directory server `ldap://my_host:3060` is added as partial read-only replica by specifying naming contexts to be replicated to directory server `ldap://my_host:3040` for which the following command is issued:

```
remtool -paddnode -v -bind my_host:3040/ods
```

The results are:

```
Directory Replication Group (DRG) details :

--- ----------------- ---------------------- ---------------------- -----
Sl  Replicaid         Directory Information   Supplier Information    Repl.
No.                                                                  Type
--- ----------------- ---------------------- ---------------------- -----
001 my_host_rid       my_host:3040             --                    RW

--- ----------------- ---------------------- ---------------------- -----
Enter consumer directory details:
Enter hostname of host running OID server    : my_host

Enter port on which OID server is listening  : 3060

Enter replication dn password                :
-------------------------------------------------------------------------
ldap://my_host:3060 [my_host_rid2] : Modifying entry orclreplicaid=my_host_
rid2,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Modifying entry orclreplicaid=my_host_
rid1,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Modifying entry ...
```

```
ldap://my_host:3040 [my_host_rid1] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Adding entry
orclagreementid=000002,orclreplicaid=my_host_rid1,cn=replication
configuration...
ldap://my_host:3040 [my_host_rid1] : Adding entry orclreplicaid=my_host_
rid2,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Adding entry cn=replication
dn,orclreplicaid=my_host_rid2,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Adding entry orclreplicaid=my_host_
rid1,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Adding entry
orclagreementid=000002,orclreplicaid=my_host_rid1,cn=replication
configuration...
ldap://my_host:3040 [my_host_rid] : Adding entry
cn=includednamingcontext000001,orclagreementid=000002,orclreplicaid=usunnae07_
prep,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Adding entry
cn=includednamingcontext000001,orclagreementid=000002,orclreplicaid=usunnae07_
prep,cn=replication configuration...
-------------------------------------------------------------------------------
Replica ldap://my_host:3060(my_host_rid2) has been added to this DRG.
-------------------------------------------------------------------------------
Directory Replication Group (DRG) details :

--- ------------------ ---------------------- ---------------------- -----
Sl  Replicaid          Directory Information   Supplier Information    Repl.
No.                                                                   Type
--- ------------------ ---------------------- ---------------------- -----
001 my_host_rid1        my_host:3040            --                     RW

002 my_host_rid2        my_host:3060            my_host_rid1           RO

--- ------------------ ---------------------- ---------------------- -----
Replica ldap://my_host:3060 (my_host_rem2) can be made partial replica by
specifying naming contexts to be replicated.

-------------------------------------------------------------------------------
List of available naming contexts in supplier replica ldap://my_host:3040 (my_
host_rid1) :

    1. * [replicate whole directory]
    2. dc=com
    3. dc=org
    4. dc=net
    5. dc=edu
```

```
Enter naming context (e-end, q-quit) : dc=org

Enter naming context (e-end, q-quit) : dc=edu

Enter naming context (e-end, q-quit) : e

Following naming contexts will be included for replication:
-------------------------------------------------------------------------------
    1. dc=org
    2. dc=edu
Do you want to continue? [y/n] : y

ldap://my_host:3040 [my_host_rid1] : Adding entry
cn=includednamingcontext000002,orclagreementid=000002,orclreplicaid=my_host_
rid,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Adding entry
cn=includednamingcontext000002,orclagreementid=000002,orclreplicaid=my_host_
rid,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Adding entry
cn=includednamingcontext000003,orclagreementid=000002,orclreplicaid=my_host_
rid,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Adding entry
cn=includednamingcontext000003,orclagreementid=000002,orclreplicaid=my_host_
rid,cn=replication configuration...

-------------------------------------------------------------------------------
Selected naming contexts have been included for replication.
-------------------------------------------------------------------------------
```

Example #2: -PADDNODE Option

In this example, directory server `ldap://my_host:3060` is added as partial replica to directory server `ldap://my_host:3040`, which is part of the DRG consisting of `ldap://my_host:3040` and `ldap://my_host:3080` that uses LDAP based replication. The user can connect to either `my_host:3040` or `my_host:3080` and add a consumer replica to `my_host:3040`.

In this example, the following command is issued:

```
remtool -paddnode -v -bind my_host:3040/ods
```

The results are:

```
Directory Replication Group (DRG) details :

--- ----------------- ---------------------- ---------------------- -----
```

```
Sl  Replicaid          Directory Information   Supplier Information    Repl.
No.                                                                    Type
--- ------------------ ---------------------- ---------------------- -----
001 my_host_rid1       my_host:3040            --                     RW

002 my_host_rid3       my_host:3080            my_host_rid1           RO

--- ------------------ ---------------------- ---------------------- -----
Enter consumer directory details:
Enter hostname of host running OID server    : my_host

Enter port on which OID server is listening  : 3060

Enter replication dn password                :
Enter replicaid of the supplier              : my_host_rid1
------------------------------------------------------------------------------
ldap://my_host:3060 [my_host_r[my_host_rid1]id2] : Modifying entry
orclreplicaid=my_host_rid2,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Modifying entry orclreplicaid=my_host_
rem,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Adding entry
orclagreementid=000003,orclreplicaid=my_host_rid,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Adding entry orclreplicaid=my_host_
rem2,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Adding entry cn=replication
dn,orclreplicaid=my_host_rem2,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Adding entry orclreplicaid=my_host_
rem2,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Adding entry cn=replication
dn,orclreplicaid=my_host_rem2,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Adding entry orclreplicaid=my_host_
rem,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Adding entry
orclagreementid=000002,orclreplicaid=my_host_rem,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Adding entry
orclagreementid=000003,orclreplicaid=my_host_rid,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Adding entry cn=replication
dn,orclreplicaid=my_host_rid,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Adding entry orclreplicaid=my_host_
rem3,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Adding entry cn=replication
dn,orclreplicaid=my_host_rid3,cn=replication configuration...
```

```
ldap://my_host:3080 [my_host_rid3] : Adding entry
orclagreementid=000003,orclreplicaid=my_host_rid,cn=replication configuration...
------------------------------------------------------------------------------
Replica ldap://my_host:3060(my_host_rem2) has been added to this DRG.
------------------------------------------------------------------------------
Directory Replication Group (DRG) details :

--- ----------------- ---------------------- ---------------------- -----
Sl  Replicaid         Directory Information   Supplier Information    Repl.
No.                                                                  Type
--- ----------------- ---------------------- ---------------------- -----
001 my_host_rid1        my_host:3040              --                    RW

002 my_host_rid2       my_host:3060               my_host_rid1          RO

003 my_host_rid3       my_host:3080               my_host_rid1          RO

--- ----------------- ---------------------- ---------------------- -----
Replica ldap://my_host:3060 (my_host_rid2) can be made partial replica by
specifying naming contexts to be replicated.
------------------------------------------------------------------------------

List of available naming contexts in supplier replica ldap://my_host:3040 (my_
host_rid1) :

    1. *  [replicate whole directory]
Enter naming context (e-end, q-quit) : e

------------------------------------------------------------------------------

------------------------------------------------------------------------------
```

Example #3:-PADDNODE Option

In this example, OID server ldap://my_host:3080 is added as partial replica to OID server ldap://my_host:3040 that is part of the DRG consisting of ldap://my_host:3040 and ldap://my_host:3060 that uses ASR based replication. The user must connect to my_host:3040 to add a consumer replica to my_host:3040 in this case. In this example, the following command is issued:

```
remtool -paddnode -v -bind my_host:3040/ods
```

The results are:

```
Directory Replication Group (DRG) details :
```

```
--- ------------------ ---------------------- ---------------------- -----
Sl  Replicaid          Directory Information   Supplier Information    Repl.
No.                                                                   Type
--- ------------------ ---------------------- ---------------------- -----
001 my_host_rid1       my_host:3040           my_host_rid2           RW

002 my_host_rid2       --                     my_host_rid1           RW

--- ------------------ ---------------------- ---------------------- -----
Enter consumer directory details:
Enter hostname of host running OID server    : my_host

Enter port on which OID server is listening  : 3080

Enter replication dn password                :
Enter replicaid of the supplier              : my_host_rid1

------------------------------------------------------------------------------
ldap://my_host:3080 [my_host_rid3] : Modifying entry orclreplicaid=my_host_
rem3,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Modifying entry orclreplicaid=my_host_
rem,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Adding entry
orclagreementid=000002,orclreplicaid=my_host_rem,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Adding entry orclreplicaid=my_host_
rem3,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Adding entry cn=replication
dn,orclreplicaid=my_host_rem3,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Adding entry orclreplicaid=my_host_
rem,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Adding entry
orclagreementid=000002,orclreplicaid=my_host_rem,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Adding entry cn=replication
dn,orclreplicaid=my_host_rem,cn=replication configuration...
------------------------------------------------------------------------------
Replica ldap://my_host:3080(my_host_rem3) has been added to this DRG.
------------------------------------------------------------------------------
Directory Replication Group (DRG) details :

--- ------------------ ---------------------- ---------------------- -----
Sl  Replicaid          Directory Information   Supplier Information    Repl.
No.                                                                   Type
```

```
--- ----------------- ---------------------- ---------------------- -----
001 my_host_rid1      my_host:3040           my_host_rid2           RW

002 my_host_rid2      --                     my_host_rid1           RW

003 my_host_rid3      my_host:3080           my_host_rid1           RO

--- ----------------- ---------------------- ---------------------- -----
Replica ldap://my_host:3080 (my_host_rid3) can be made partial replica by
specifying naming contexts to be replicated.
Do you want to continue? [y/n] : y

-------------------------------------------------------------------------------
List of available naming contexts in supplier replica ldap://my_host:3040 (my_
host_rid1) :

    1. * [replicate whole directory]
    2. dc=com
    3. dc=org
    4. dc=net
    5. dc=edu
Enter naming context (e-end, q-quit) : *

Enter naming context (e-end, q-quit) : e

Following naming contexts will be included for replication:
-------------------------------------------------------------------------------
    1. *
Do you want to continue? [y/n] : y

ldap://my_host:3040 [my_host_rid] : Adding entry
cn=includednamingcontext000002,orclagreementid=000002,orclreplicaid=my_host_
rid,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Adding entry
cn=includednamingcontext000002,orclagreementid=000002,orclreplicaid=my_host_
rid,cn=replication configuration...
-------------------------------------------------------------------------------
Selected naming contexts have been included for replication.
-------------------------------------------------------------------------------
```

-PDELNODE Option

The syntax is:

```
remtool -pdelnode [-v] [-bind <hostname>:<port#>/<repl_dn_password>]
```

Usage Notes for the -PDELNODE Option

1. This option can be used to delete a read-only or read-only partial replica from a DRG.
2. This option cannot be used to delete a ASR based replica. -delnode option has to be used.

Example 1: -PDELNODE Option

In this example, replica `ldap://my_host:3080` is removed from the DRG created as shown in Example 3 of -PADDNODE option. This DRG consists of 3 replicas - `ldap://my_host:3040`, `ldap://my_host:3060`, `ldap://my_host:3080` - of which `ldap://my_host:3040` and `ldap://my_host:3060` uses ASR based replication and `ldap:my_host:3040` and `ldap://my_host:3080` uses LDAP based replication. To delete replica `ldap://my_host:3080`, user has to give bind details of either `ldap://my_host:3040` or `ldap://my_host:3080`.

> **Note:** A user cannot delete the replica `ldap://my_host:3080` by giving bind details of `ldap://my_host:3060`, although binding to it gives details of all replicas .

In this example, the following command is issued:

```
remtool -pdelnode -v -bind my_host:3040/ods
------------------------------------------------------------------------------
Directory Replication Group (DRG) details :
--- ----------------- ---------------------- ---------------------- -----
Sl   Replicaid         Directory Information    Supplier Information   Repl.
No.                                                                   Type
--- ----------------- ---------------------- ---------------------- -----
001  my_host_rid1       my_host:3040           my_host_rid2           RW

002  my_host_rid2       --                     my_host_rid1           RW

003  my_host_rid3       my_host:3080           my_host_rid1           RO

--- ----------------- ---------------------- ---------------------- -----
Enter replicaid of the replica to be deleted : my_host_rid3

------------------------------------------------------------------------------
```

```
ldap://my_host:3040 [my_host_rid1] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Deleting entry
orclagreementid=000002,orclreplicaid=my_host_rid1,cn=replication
configuration...
ldap://my_host:3040 [my_host_rid1] : Deleting entry orclreplicaid=my_host_
rem3,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Modifying entry orclreplicaid=my_host_
rem3,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Modifying entry ...
ldap://my_host:3080 [my_host_rid3] : Deleting entry orclreplicaid=my_host_
rem,cn=replication configuration...
-------------------------------------------------------------------------------
Replica ldap://my_host:3080(my_host_rid3) has been deleted from this DRG.
-------------------------------------------------------------------------------
Directory Replication Group (DRG) details :

--- ------------------ ---------------------- ---------------------- -----
Sl  Replicaid          Directory Information   Supplier Information    Repl.
No.                                                                   Type
--- ------------------ ---------------------- ---------------------- -----
001 my_host_rid1       my_host:3040           my_host_rid2            RW

002 my_host_rid2       --                     my_host_rid1            RW

--- ------------------ ---------------------- ---------------------- -----
```

Example #2: -PDELNODE Option

In this example, a replica is deleted from a DRG consisting of three replicas. All three replicas use LDAP-based replication. Required replica can be deleted by binding to any of these three replicas.

In this example, the following command is issued:

```
remtool -pdelnode -v -bind my_host:3040/ods
-------------------------------------------------------------------------------
Directory Replication Group (DRG) details :

--- ------------------ ---------------------- ---------------------- -----
Sl  Replicaid          Directory Information   Supplier Information    Repl.
No.                                                                   Type
--- ------------------ ---------------------- ---------------------- -----
001 my_host_rid1       my_host:3040           --                      RW

002 my_host_rid3       my_host:3080           my_host_rid1            RO
```

```
003 my_host_rid2       my_host:3060            my_host_rid1             RO

--- ----------------- ---------------------- ---------------------- -----
Enter replicaid of the replica to be deleted : my_host_rid3

------------------------------------------------------------------------------
ldap://my_host:3040 [my_host_rid1] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Deleting entry
orclagreementid=000003,orclreplicaid=my_host_rem,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Deleting entry orclreplicaid=my_host_
rem3,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Deleting entry
orclagreementid=000003,orclreplicaid=my_host_rem,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Deleting entry orclreplicaid=my_host_
rem3,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Modifying entry orclreplicaid=my_host_
rem3,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Modifying entry ...
ldap://my_host:3080 [my_host_rid3] : Deleting entry orclreplicaid=my_host_
rem,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Deleting entry orclreplicaid=my_host_
rem2,cn=replication configuration...
------------------------------------------------------------------------------
Replica ldap://my_host:3080(my_host_rid3) has been deleted from this DRG.
------------------------------------------------------------------------------
Directory Replication Group (DRG) details :

--- ----------------- ---------------------- ---------------------- -----
Sl  Replicaid         Directory Information   Supplier Information    Repl.
No.                                                                  Type
--- ----------------- ---------------------- ---------------------- -----
001 my_host_rid1      my_host:3040            --                     RW

002 my_host_rid2      my_host:3060            my_host_rid            RO

--- ----------------- ---------------------- ---------------------- -----
```

-PCHGPWD Option

The syntax is:

```
remtool -pchgpwd [-v] [-bind <hostname>:<port>/<replication_dn_password>]
```

Usage Notes for the -PCHGPWD Option

1. This option is used to change password of replication DN.

2. The replication DN of Oracle Internet Directory server identified by "-bind" option will be changed.
3. Password of replication DN of the identified replica will be changed both in Oracle Internet Directory repository and in wallet.
4. If the replica is taking part in replication, then password will be changed in other replicas for the local replica's replication DN. Note that, unlike ASR based replication, the replication DN password of each replica can be different from others.
5. This option has to be executed in the host, where Oracle Internet Directory server whose replication DN password has to be changed is running. This is mandatory as password in wallet must also to be changed. Otherwise remtool will report an error as shown in example #2.

Example 1: -PCHGPWD Option

In this example, the password of replica `ldap://my_host:3040/ods` is changed for which the following command is issued:

```
remtool -pchgpwd -v -bind my_host:3040/ods
```

The results are:

```
Directory Replication Group (DRG) details :

--- ----------------- ---------------------- ---------------------- -----
Sl  Replicaid         Directory Information   Supplier Information    Repl.
No.                                                                  Type
--- ----------------- ---------------------- ---------------------- -----
001 my_host_rid1      my_host:3040            --                     RW

002 my_host_rid3      my_host:3080            my_host_rid1           RO

--- ----------------- ---------------------- ---------------------- -----
------------------------------------------------------------------------------
Replication DN password of ldap://my_host:3040 (my_host_rem) associated with
database 'rid' will be changed.
Do you want to continue? [y/n] : y

Enter new password of replication DN        :
Reenter new password of replication DN      :
------------------------------------------------------------------------------
ldap://my_host:3040 [my_host_rid1] : Modifying entry cn=replication
dn,orclreplicaid=my_host_rem,cn=replication configuration...
```

```
ldap://my_host:3080 [my_host_rid3] : Modifying entry cn=replication
dn,orclreplicaid=my_host_rem,cn=replication configuration...
-------------------------------------------------------------------------------
Password has been changed.
-------------------------------------------------------------------------------
```

Example 2: -PCHGPWD Option

In this example, user tries to change the password of replica `my_host:3040` from a different host for which the following command is issued:

```
remtool -pchgpwd -v -bind my_host:3040/ods
```

The results are:

```
Directory Replication Group (DRG) details :

--- ----------------- ---------------------- ---------------------- -----
Sl  Replicaid         Directory Information   Supplier Information    Repl.
No.                                                                  Type
--- ----------------- ---------------------- ---------------------- -----
001 my_host_rid1      my_host:3040           --                     RW

002 my_host_rid3      my_host:3080           my_host_rid1           RO

--- ----------------- ---------------------- ---------------------- -----
-------------------------------------------------------------------------------
Replication DN password of ldap://my_host:3040 (my_host_rid1) associated with
database 'rid1' will be changed.
Do you want to continue? [y/n] : y

Enter new password of replication DN        :
Reenter new password of replication DN      :
-------------------------------------------------------------------------------
ldap://my_host:3040 : Invoke the remtool at host my_host to change the password
of ldap://my_host:3040 replica.
-------------------------------------------------------------------------------
Error occurred while changing password of replica ldap://my_host:3040(my_host_
rid1).
ldap://my_host:3040 : Invoke the remtool at host my_host to change the password
of ldap://my_host:3040 replica.
```

-PCLEANUP Option

```
The syntax is:
remtool -pcleanup -v -bind my_host:3040/ods
```

Usage Notes for the -PCLEANUP Option

1. This option can be used to clean up LDAP based replication setup.
2. This option can be used to clean up a replica which has incomplete or flawed LDAP based replication setup. In case of incomplete or flawed LDAP-based replication setup, the Replication Environment Management Tool cleans up only the replica identified by the `-bind` option. If replication configuration information is corrupted, or the replication DN entry is not available, then it prompts for the super user DN and password.
3. This option can be used only to clean up LDAP based replication setup and not ASR based replication setup.

Example 1: -PCLEANUP Option

In this example, the replication setup of a DRG that has 3 replicas taking part in LDAP based replication.

In this example, the following command is issued:

```
remtool -pcleanup -v -bind my_host:3040/ods
```

The results are:

```
Directory Replication Group (DRG) details :

--- ----------------- ---------------------- ---------------------- -----
Sl  Replicaid         Directory Information   Supplier Information    Repl.
No.                                                                  Type
--- ----------------- ---------------------- ---------------------- -----
001 my_host_rid1      my_host:3040             --                     RW

002 my_host_rid3      my_host:3080            my_host_rid1            RO

003 my_host_rid2      my_host:3060            my_host_rid1            RO

--- ----------------- ---------------------- ---------------------- -----
DRG identified by replica ldap://my_host:3040 (my_host_rid1) will be cleaned up.
Do you want to continue? [y/n] : y

-------------------------------------------------------------------------------
ldap://my_host:3040 [my_host_rid1] : Modifying entry orclreplicaid=my_host_
rem,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Deleting entry
```

```
orclagreementid=000002,orclreplicaid=my_host_rem,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Deleting entry
orclagreementid=000003,orclreplicaid=my_host_rem,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Deleting entry orclreplicaid=my_host_
rem3,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Deleting entry orclreplicaid=my_host_
rem2,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Modifying entry orclreplicaid=my_host_
rem3,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Modifying entry ...
ldap://my_host:3080 [my_host_rid3] : Modifying entry ...
ldap://my_host:3080 [my_host_rid3] : Deleting entry orclreplicaid=my_host_
rem,cn=replication configuration...
ldap://my_host:3080 [my_host_rid3] : Deleting entry orclreplicaid=my_host_
rem2,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Modifying entry orclreplicaid=my_host_
rem2,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Modifying entry ...
ldap://my_host:3060 [my_host_rid2] : Modifying entry ...
ldap://my_host:3060 [my_host_rid2] : Deleting entry orclreplicaid=my_host_
rem3,cn=replication configuration...
ldap://my_host:3060 [my_host_rid2] : Deleting entry cn=replication
dn,orclreplicaid=my_host_rem3,cn=replication configuration...
-------------------------------------------------------------------------------
Replica ldap://my_host:3040(my_host_rid1) has been cleaned up.
-------------------------------------------------------------------------------
```

Example 2: -PCLEANUP Option

This example shows how -pcleanup option can be used to clean up flawed LDAP based replication setup.

Step 1: First replica `ldap://my_host:3040` is added to `ldap://my_host:3060`.While replication setup is in progress, error occurs which results in flawed setup.

```
remtool -paddnode -v -bind my_host:3040/ods

Directory Replication Group (DRG) details :

--- ------------------ ---------------------- ---------------------- -----
Sl  Replicaid          Directory Information   Supplier Information    Repl.
No.                                                                   Type
--- ------------------ ---------------------- ---------------------- -----
001 my_host_rid1       my_host:3040           --                     RW
```

```
--- ----------------- ---------------------- ---------------------- -----
Enter consumer directory details:
Enter hostname of host running OID server    : my_host
Enter port on which OID server is listening  : 3060
Enter replication dn password                :
------------------------------------------------------------------------
------------------------------------------------------------------------
Error occurred while adding partial replica ldap://my_host:3060.
ldap://my_host:3060 : Failed to add entry orclreplicaid=my_host_
rid1,cn=replication configuration.
DSA is unwilling to perform

ldap://my_host:3060 : Failed to read replication configuration information.
```

Step 2: Again add ldap://my_host:3060 to ldap://my_host:3040, which results in error.

```
remtool -paddnode -v -bind my_host:3040/ods
ldap://my_host:3060 : Failed to read replication configuration information.
```

Step 3: As there was error in above paddnode procedure, no new node can be added. Hence call -pcleanup to clean the setup. After cleanup is complete, -paddnode can be invoked again to add a new replica.

```
remtool -pcleanup -v -bind my_host:3040/ods
ldap://my_host:3060 : Failed to read replication configuration information.
Error occurred while getting replication configuration information.
This tool will try to rectify the problem if super user DN and password are
provided.
Do you want to continue? [y/n] : y

Enter superuser DN                          : cn=orcladmin

Enter superuser password                    :
Enter new password of replication DN        :
Reenter new password of replication DN      :
------------------------------------------------------------------------
Directory Replication Group (DRG) details :

--- ----------------- ---------------------- ---------------------- -----
Sl  Replicaid         Directory Information   Supplier Information    Repl.
No.                                                                  Type
--- ----------------- ---------------------- ---------------------- -----
001 my_host_rid1      my_host:3040           --                     RW
```

```
002 my_host_rid2       my_host:3060            my_host_rid1            RO

--- ----------------- ---------------------- ---------------------- -----
DRG identified by replica ldap://my_host:3040 (my_host_rem) will be cleaned up.
Do you want to continue? [y/n] : y

-------------------------------------------------------------------------------
ldap://my_host:3040 [my_host_rid1] : Modifying entry orclreplicaid=my_host_
rem,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Modifying entry ...
ldap://my_host:3040 [my_host_rid1] : Deleting entry
orclagreementid=000002,orclreplicaid=my_host_rem,cn=replication configuration...
ldap://my_host:3040 [my_host_rid1] : Deleting entry orclreplicaid=my_host_
rem2,cn=replication configuration...
-------------------------------------------------------------------------------
Replica ldap://my_host:3040(my_host_rem) has been cleaned up.
-------------------------------------------------------------------------------
```

-PRESETPWD Option

The syntax is:

```
remtool -presetpwd -v -bind my_host:3040/ods
```

Usage Notes for the -PRESETPWD Option

1. This option is used for resetting replication DN password.
2. To reset the replication DN password, Superuser DN and password are required.
3. This will just reset the replication DN password in wallet and in the directory.
4. This will not reset the password in any other directories of the DRG of which this directory is part.

Example: -PRESETPWD Option

In this example, the following command is issued to reset the password of replica `my_host:3040`:

```
remtool -presetpwd -v -bind my_host:3040/ods
```

The results are:

```
Enter superuser DN                          : cn=orcladmin
```

```
Enter superuser password                    :
-------------------------------------------------------------------------------
Replication DN password of ldap://my_host:3040 (my_host_rem) associated with
database 'rid1' will be reset.
Do you want to continue? [y/n] : y

Enter new password of replication DN        :
Reenter new password of replication DN      :
-------------------------------------------------------------------------------
ldap://my_host:3040 [my_host_rid1] : Modifying entry cn=replication
dn,orclreplicaid=my_host_rid1,cn=replication configuration...
-------------------------------------------------------------------------------
Password has been changed.
-------------------------------------------------------------------------------
```

-PCHGWALPWD Option

The syntax is:

```
remtool -pchgwalpwd -v -bind my_host:3040/ods
```

Usage Notes for the -PCHGWALPWD Option

1. This option is used to change only the wallet password.
2. This will set the wallet password to replication DN password stored in Oracle Internet Directory repository.
3. Bind details must be that of the directory whose wallet password is to be changed.
4. This option is useful in RAC environment.

Example: -PCHGWALPWD Option

In this example, password of replication DN of replica `ldap://my_host:3040` is set to that of in Oracle Internet Directory repository in the wallet for which the following command is issued:

```
remtool -pchgwalpwd -v -bind my_host:3040/ods
```

The results are:

```
Directory Replication Group (DRG) details :

--- ----------------- ---------------------- ---------------------- -----
Sl  Replicaid         Directory Information   Supplier Information    Repl.
```

```
No.                                                                    Type
--- ----------------- ---------------------- ---------------------- -----
001 my_host_rid1      my_host:3040           --                     RW

002 my_host_rid3      my_host:3080           my_host_rid1           RO

--- ----------------- ---------------------- ---------------------- -----
-------------------------------------------------------------------------------
Replication DN password of ldap://my_host:3040 (my_host_rid1) associated with
database 'rid' will be set in wallet.
Do you want to continue? [y/n] : y
```

Password has been changed.

Oracle Directory Integration and Provisioning Platform Command-Line Tools Syntax

This section contains these topics:

- The Directory Integration and Provisioning Assistant
- The ldapUploadAgentFile.sh Tool Syntax
- The ldapCreateConn.sh Tool Syntax
- The ldapDeleteConn.sh Tool Syntax
- The StopOdiServer.sh Tool Syntax
- The schemasync Tool Syntax
- The Oracle Directory Integration and Provisioning Server Registration Tool (odisrvreg)
- The Provisioning Subscription Tool (oidprovtool) Syntax

The Directory Integration and Provisioning Assistant

Table A–26 lists the tasks you can perform by using the Directory Integration and Provisioning Assistant and the corresponding commands. It also points you to instructions for performing each task.

Table A–26 Summary of Functionality of the Directory Integration and Provisioning Assistant

Tasks	Commands	More Information

Create, modify, or delete a synchronization profile	`createprofile`
	`modifyprofile`
	`deleteprofile`
		"Creating, Modifying, and Deleting Synchronization Profiles" on page A-109

See all the profile names in Oracle Internet Directory	`listprofiles`	"Listing All Synchronization Profiles in Oracle Internet Directory" on page A-116
See the details of a specific profile	`showprofile`	"Viewing the Details of a Specific Synchronization Profile" on page A-116
Make Oracle Internet Directory and the connected directory identical before beginning synchronization	`bootstrap`	"Bootstrapping a Directory by Using the Directory Integration and Provisioning Assistant" on page A-111
Set the wallet password that the Oracle directory integration and provisioning server later uses to connect to Oracle Internet Directory	`wpasswd`	"Setting the Wallet Password for the Oracle Directory Integration and Provisioning Server" on page A-117
Reset the password of the administrator of the Oracle Directory Integration Platform	`chgpasswd`	"Changing the Password of the Administrator of the Oracle Directory Integration and Provisioning Platform" on page A-116
Move integration profiles from one identity management node to another	`reassociate`	"Moving an Integration Profile to a Different Identity Management Node" on page A-117

The command-line interface for the Directory Integration and Provisioning Assistant is:

dipassistant *command* [-help]

command := *Directory Integration and Provisioning Assistant command*

```
Directory Integration and Provisioning Assistant command :=
        createprofile [cp]
        | modifyprofile [mp]
        | deleteprofile [dp]
        | listprofiles[lsprof]
        | showprofile[sp]
        | bootstrap [bs]
        | wpasswd [wp]
```

```
        | chgpasswd [cpw]
        | reassociate [rs]
```

For help on a particular command, enter:

```
dipassistant command -help
```

Creating, Modifying, and Deleting Synchronization Profiles

The syntax for creating, modifying, or deleting synchronization profiles by using the Directory Integration and Provisioning Assistant is:

```
dipassistant createprofile | modifyprofile | deleteprofile
[-host host name] [-port port number] [-dn bind_DN] [-passwd password]
{-file file name | -profile profile name } [propName1=value]
[propName2=value]... [-configset configset_number]
```

For example:

```
dipassistant createprofile -host myhost -port 3060 -passwd xxxx
-file import.profile -configset 1

dipassistant modifyprofile -host myhost -port 3060 -passwd xxxx
-file import.profile -dn xxxx -passwd xxxx -profile myprofile
[propName1=value]
[propName2=value]...

dipassistant deleteprofile -profile myprofile [-host myhost] [-port 3060] [-dn
xxxx] [-passwd xxxx] [-configset 1]
```

Table A–27 on page A-109 describes the parameters for creating, modifying, and deleting synchronization profiles by using the Directory Integration and Provisioning Assistant.

Table A–27 Parameters for Creating, Modifying, and Deleting Synchronization Profiles by Using the Directory Integration and Provisioning Assistant

Parameter	Description
-host	Host where Oracle Internet Directory is running. The default value is the name of the local host.
-port	Port at which Oracle Internet Directory was started. The default is 389.
-dn	The Bind DN to be used in identifying to the directory. The default value is the DN of the Oracle Directory Integration and Provisioning platform administrator.

Table A–27 (Cont.) Parameters for Creating, Modifying, and Deleting Synchronization Profiles by Using the Directory Integration and Provisioning Assistant

Parameter	Description
`-passwd`	The password of the bind DN to be used while binding to the directory.
`-file`	The file containing all the profile parameters. **See Also:** Table A–28 on page A-110 for a list of parameters and their description
`-configset`	Number of the configuration set entry with which the profile needs to be associated
`-profile`	Profile that needs to modified

The properties expected by `createprofile` and `modifyprofile` commands are described in Table A–28. When modifying an already existing profile, no defaults are assumed. Only those attributes specified in the file are changed.

Table A–28 Properties Expected by createprofile and modifyprofile Commands

Parameter	Description	Default
`odip.profile.name`	Name of the profile	-
`odip.profile.password`	Password for accessing this profile	-
`odip.profile.status`	Either `DISABLE` or `ENABLE`	`DISABLE`
`odip.profile.syncmode`	Direction of synchronization. When the changes are propagated from the third party to Oracle Internet Directory, the synchronization mode is IMPORT. When the changes are propagated to the third party directory, the synchronization mode is EXPORT.	IMPORT
`odip.profile.retry`	Maximum number of times this profile should be executed in the case of an error before the integration server gives up	4
`odip.profile.schedinterval`	Interval between successive executions of this profile by the integration server. If the previous execution has not completed then the next execution will not resume until it completes.	1 Minute
`odip.profile.agentexecommand`	In the case of a NON-LDAP interface, the command to produce the information in LDIF format	-
`odip.profile.condirurl`	Location of third-party directory [`hostname:port`]	-

Table A–28 (Cont.) Properties Expected by createprofile and modifyprofile Commands

Parameter	Description	Default
odip.profile.condiraccount	DN or user name used to connect to the third party directory.	-
odip.profile.condirpassword	Password used for identification to the third-party directory.	-
odip.profile.interface	Indicator as to whether the LDAP or LDIF or DB or TAGGED format is to be used for data exchange	LDAP
odip.profile.configfile	Name of the file that contains the additional profile-specific information to be used for execution	-
odip.profile.mapfile	Name of the file that contains the mapping rules	-
odip.profile.condirfilter	Filter that needs to be applied to the changes read from the connected directory before importing to Oracle Internet Directory	-
odip.profile.oidfilter	Filter that needs to be applied to the changes that are read from the Oracle Internet Directory before exporting to the connected directory	-
odip.profile.lastchgnum	Last applied change number. In the case of an export profile this number refers to Oracle Internet Directory's last applied change number However, n the case of the import profile, this number refers to the last applied change number in the connected directory	-

Bootstrapping a Directory by Using the Directory Integration and Provisioning Assistant

The command-line interface to the bootstrap command is:

```
dipassistant bootstrap { -profile profile_name [-host host_name] [-port port_
number] -dn bind_DN [-passwd password] [-log log_file] [-logseverity severity]
[-trace trace_file] [-tracelevel trace_level] [-loadparallelism <#nThrs>]
[-loadretry <retryCnt>] | -cfg file_name }
```

For example, either:

```
dipassistant bs -cfg bootstrap cfg
```

or

```
dipassistant bs -host myhost -port 3060 -dn cn=orcladmin -password xxxx -profile
iPlanetProfile
```

Table A–29 Parameters of a deleteprofile Command

Parameter	Description
`-cfg`	A configuration file containing all the parameters required for performing the bootstrapping. **See Also:** Table A–30 on page A-113 for a list of parameters and their description
`-host`	Host where Oracle Internet Directory is running
`-port`	Port at which Oracle Internet Directory was started
`-dn`	The Bind Dn to be used in identifying to the directory
`-password`	The password of the Bind DN to be used while binding to the directory
`-profile`	The profile name.
`-log`	Log file. If this parameter is not specified, then, by default, the log information is written to `OH/ldap/odi/bootstrap.log`
`-logseverity`	Log severity 1 - 15. 1 – INFO, 2 – WARNING, 3 – DEBUG, 4 – ERROR. Or any combination of these. If not specified, then INFO and ERROR messages alone will be logged.
`-trace`	Trace file for debugging purpose
`-trace level`	Trace level
`-loadRetry`	When the loading to the destination fails, the number of times the retry should be made before marking the entry as bad entry
`-loadparallelism`	Indicator that loading to Oracle Internet Directory is to take place in parallel by using multiple threads. For example, `-loadparallelism 5` means that 5 threads are to be created, each of which tries to load the entries in parallel to Oracle Internet Directory.

Properties Expected by the Bootstrapping Command

Table A–30 Bootstrapping Properties

Property	Description	Mandatory	Default
`odip.bootstrap.srctype`	Indicator of whether source of the bootstrapping is LDAP or LDIF. Valid values are either `LDAP` or `LDIF`.	Yes	-
`odip.bootstrap.desttype`	Indicator of whether destination of the bootstrapping is LDAP or LDIF. Valid values are either `LDAP` or `LDIF`.	Yes	-
`odip.bootstrap.srcurl`	In the case of LDAP source type, location of the source directory. In the case of LDIF, the location of the LDIF file. **Note:** For LDAP, the expected format is `host[:port]`. For LDIF, the expected format is the absolute path of the file.	Yes	-
`odip.bootstrap.desturl`	In the case of LDAP, location of the destination directory. In the case of LDIF, the location of the LDIF file. **Note:** For LDAP, the expected format is `host[:port]`. For LDIF, the expected format is the absolute path of the file.	Yes	-
`odip.bootstrap.srcsslmode`	Indicator of whether SSL-based authentication must be used to connect to the source of the bootstrapping. A value of `TRUE` indicates that SSL-based authentication must be used.	No	`FALSE`

Table A–30 (Cont.) Bootstrapping Properties

Property	Description	Mandatory	Default
`odip.bootstrap.destsslmode`	Indicator of whether SSL-based authentication must be used to connect to the destination of the bootstrapping. `TRUE` indicates that SSL-based authentication must be used. **Note:** In the case of LDIF, this parameter is meaningless.	No	`FALSE`
`odip.bootstrap.srcdn`	Supplement to the source URL. In the case of LDIF binding, this parameter is meaningless. However in the case of LDAP, this parameter specifies the Bind DN.	Only in the case of LDAP	-
`odip.bootstrap.destdn`	Supplement to the destination URL. In the case of LDIF binding, this parameter is meaningless. However in the case of LDAP, this parameter specifies the Bind DN.	Only in the case of LDAP	-
`odip.bootstrap.srcpasswd`	Bind password to the source. In the case of LDAP binding, this is used as security. Oracle Corporation recommends that you not specify the password in this file.	No	-
`odip.bootstrap.destpasswd`	Bind password. In the case of LDAP binding, this is used as security credential. Oracle Corporation recommends that you not specify the password in this file.	No	-

Table A–30 (Cont.) Bootstrapping Properties

Property	Description	Mandatory	Default
`odip.bootstrap.mapfile`	Location of the map file that contains the attribute and domain mappings.	No	-
`odip.bootstrap.logfile`	Location of the log file. If this file already exists then it will be appended. The default log file is `bootstrap.log` created under `$`*`ORACLE_HOME`*`/ldap/odi/log directory.`	No	The file `bootstrap.log` created under the directory `$`*`ORACLE_HOME`*`/ldap/odi/`
`odip.bootstrap.logseverity`	Type of log messages that needs to be logged. INFO – 1 WARNING - 2 DEBUG – 4 ERROR - 8 **Note:** A combination of these types can also be given. For example, if you are interested only in WARNING and ERROR message, then specify a value of 8+2—that is, 10. Similarly, for all types of message, use 1 + 2 + 4 + 8 = 15	No	1 + 8 = 9
`odip.bootstrap.loadparallelism`	Numeric value indicating the number of writer threads used to load the processed data to the destination	No	1-
`odip.bootstrap.loadretry`	In the event of a failure to load an entry, indicator of how many times to retry	No	5
`odip.bootstrap.trcfile`	Location of the trace file. If this file already exists, then it is overwritten.	No	`$`*`ORACLE_HOME`*`/ldap/odi/log/bootstrap.trc`
`odip.bootstrap.trclevel`	The tracing level	No	3

Changing the Password of the Administrator of the Oracle Directory Integration and Provisioning Platform

The default password for the `dipadmin` account is same as `ias_admin` password chosen during installation. This command lets you reset the password of `dipadmin` account. To reset that password, you must provide the security credentials of the `orcladmin` account.

For example:

```
$ dipassistant chgpasswd -passwd orcladmin password -host oid.heman.com
-port 3060
```

The Assistant then prompts for the new password as follows:

```
New Password:
Confirm Password:
```

Listing All Synchronization Profiles in Oracle Internet Directory

The listprofiles command prints a list of all the synchronization profiles in Oracle Internet Directory. For example:

```
$ dipassistant listprofiles -passwd dipadmin password -host oid.heman.com
-port 3060
```

This command prints the following sample list:

```
IplanetExport
IplanetImport
ActiveImport
ActiveExport
LdifExport
LdifImport
TaggedExport
TaggedImport
OracleHRAgent
ActiveChgImp
```

Note: The list shown here is the default set of profiles created during installation.

Viewing the Details of a Specific Synchronization Profile

The showprofile command prints the details of a specific synchronization profile For example:

```
$ dipassistant showprofile -passwd dipadmin password -host oid.heman.com
-port 3060 -profile ActiveImport
```

This command prints the following sample output:

```
odip.profile.version = 1.0
odip.profile.lastchgnum = 0
odip.profile.interface = LDAP
odip.profile.oidfilter = orclObjectGUID
odip.profile.schedinterval = 60
odip.profile.name = ActiveImport
odip.profile.syncmode = IMPORT
odip.profile.retry = 5
odip.profile.debuglevel = 0
odip.profile.status = DISABLE
```

Setting the Wallet Password for the Oracle Directory Integration and Provisioning Server

The *WPasswd* command enables you to set the wallet password that the Oracle directory integration and provisioning server later uses to connect to Oracle Internet Directory. To use this command, enter:

```
dipassistant wp
```

The Directory Integration and Provisioning Assistant prompts you to enter, and then confirm, the password.

Moving an Integration Profile to a Different Identity Management Node

You can use the Directory Integration and Provisioning Assistant to move directory integration profiles to another node and to reassociate them with it. For example, if the middle-tier components are associated with a particular Oracle Identity Management infrastructure, then all the integration profiles existing in that infrastructure node can be moved to a new infrastructure node.

Table A–31 describes the reassociation rules.

Table A–31 Scenarios for Reassociating Directory Integration Profiles

Scenario	Actions Taken
Integration profile does not exist on the second Oracle Internet Directory node	The integration profile is copied to the second Oracle Internet Directory node and is disabled after copying. It must be enabled by the application. The `lastchangenumber` attribute in the integration profile is modified to the current last change number on the second Oracle Internet Directory node.
Integration profile exists on the second Oracle Internet Directorynode	Both integration profiles are reconciled in the following manner: ■ Any new attribute in the profile on node 1 is added to the profile on node 2 ■ For existing same attributes, the values in profile on node 1 override the attributes in the profile on node 2 ■ The Profile is disabled after copying. It needs to be enabled by the application. ■ The `lastchangenumber` attribute in the integration profile is modified to the current last change number on the second Oracle Internet Directory node

The usage is as follows

```
dipassistant reassociate [-src_ldap_host <hostName>]
[-src_ldap_port <portNo>] [-src_ldap_dn <bindDn>] [-src_ldap_passwd
<password>] -dst_ldap_host <hostName> [-dst_ldap_port <portNo>]
[-dst_ldap_dn <bindDn>] [-dst_ldap_passwd <password>] [-log <logfile>]
Options:
-src_ldap_host <hostName> : Host where OID-1 runs
-src_ldap_port <portNo> : Port at which OID-1 runs
-src_ldap_dn <bindDn> : Bind Dn to connect to OID-1
-src_ldap_passwd <password> : Bind Dn password to connect to OID-1
-dst_ldap_host <hostName> : Host where OID-2 runs
-dst_ldap_port <portNo> : Port at which OID-2 runs
-dst_ldap_dn <bindDn> : Bind Dn to connect to OID-2
-dst_ldap_passwd <password> : Bind Dn password to connect to OID-2
-log <logFile> : Log file
```

Defaults:

```
src_ldap_host - localhost, src_ldap_port & dst_ldap_port - 389
src_ldap_dn & dst_ldap_dn - cn=orcladmin account
```

Examples:

```
dipassistant reassociate -src_ldap_host oid1.mycorp.com \
-dst_ldap_host oid2.mycorp.com -src_ldap_passwd xxxx \
-dst_ldap_passwd xxxx

dipassistant rs -help
```

Note if the location of the log file is not specified then by default it will be created as `$ORACLE_HOME/ldap/odi/log/reassociate.log`.

Limitations of the Directory Integration and Provisioning Assistant in Oracle Internet Directory 10*g* (9.0.4)

In this release, the Directory Integration and Provisioning Assistant does not support the following:

- SSL-based authentications to Oracle Internet Directory
- Schema synchronization
- Automatic profile creation at the end of the bootstrapping process when used with the -cfg option
- Mapping file validation
- Creation of a failed entries file

The following elements of the Directory Integration and Provisioning Assistant are untested:

- Bootstrapping of the connected directory over the SSL connection
- The use of the modifyprofile command while synchronization is happening for that profile

The bootstrapping command of the Directory Integration and Provisioning Assistant has the limitations described in Table A–32.

Table A–32 Limitations of Bootstrapping in the Directory Integration and Provisioning Assistant

Type of Bootstrapping	Limitation
LDIF-to-LDIF	None

Table A–32 (Cont.) Limitations of Bootstrapping in the Directory Integration and Provisioning Assistant

Type of Bootstrapping	Limitation
LDAP-to-LDIF	For a large number of entries, bootstrapping can fail with an error of size limit exceeded. To resolve this, the server from which you are bootstrapping should: ▪ Support paged results control (OID 1.2.840.113556.1.4.319). Currently, Microsoft Active Directory is the only LDAP directory that supports this control. ▪ Have an adequate value for the server side search size limit parameter ▪ Use the proprietary Import/Export tool, take the dump of the data, and bootstrap by using either the LDIF-to-LDIF or the LDIF-to-LDAP approach
LDIF -to-LDAP	None
LDAP-to-LDAP	Same as LDAP-to-LDIF

The ldapUploadAgentFile.sh Tool Syntax

Use `LdapUploadAgentFile.sh` to load mapping and configuration information when you are synchronizing directories.

```
ldapUploadAgentFile.sh -name  profile_name
-config configset_the_profile_is_associated_with
-LDAPhost  directory_server_host
-LDAPport  directory_server_port
-binddn  DN_that_can_modify_the_profile   >
-bindpass password_for_the_bind_DN
-attrtype  "MAP" | "ATTR"
-filename complete_path_of_file_to_be_uploaded
```

Table A–33 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`	Directory server host
`LDAPport`	Directory server port

Table A–33 (Cont.) Arguments for ldapUploadAgentFile.sh

Argument	Description
`Binddn`	Bind DN of the directory user who has access rights to modify the profile entry. `The default is cn=orcladmin`
`Bindpass`	Password corresponding to the bind DN. The default is `welcome`.
`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 path name of the file to be uploaded.

Note: Alternatively, you can use the Directory Integration and Provisioning Assistant to perform this operation. Enter either of the following:

```
dipassistant mp [options] odip.profile.mapfile=your
map file


dipassistant mp [options] odip.profile.configfile=
your configuration file
```

See Also: Chapter 33, "Oracle Directory Synchronization Service" for a description of when to use `ldapUploadAgentFile.sh`

The ldapCreateConn.sh Tool 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 configset_to_associate_with ] \
        [ -LDAPhost directory_server_host ]
```

```
[ -LDAPport directory_server_port ] \
[ -binddn DN_of_super_user] \
[ -bindpass   Bind_password ] \                                    [
[-retry   maximum_retry_count_on_synchronization_errors ] \
[ -poll polling_interval_for_synchronization ] \
[ -host host_on_which_to_run_agent ] \
[ -conndirurl  connected_directory_URL ] \
[ -conndiracct connected_directory_account_information ] \
[ -conndirpwd connected_directory_account_password ] \
[ -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–34 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	Directory server host. The default is the current host.
LDAPport	Directory 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 when 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

Table A–34 (Cont.) Arguments for Registering a Partner Agent by Using

Argument	Description
`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.

Note: Alternatively, you can use the `createprofile` option of the Directory Integration and Provisioning Assistant to perform this operation.

The ldapDeleteConn.sh Tool Syntax

You can deregister a synchronization profile by using the command-line tool ldapDeleteConn . sh. This tool is in the directory $*ORACLE_HOME*/ldap/admin/.

The syntax is:

```
ldapdeleteConn.sh [ -name Profile_Name ]
 -LDAPhost <LDAP server host> (default is local host)]
            [ -LDAPport directory_server_port> (default 389)]
            [ -binddn SuperUserDN (default cn=orcladmin ) ]
            [ -bindpass   password (default=welcome) ]
            [ -config configset_associated_with_agent ]
            [ -U <SSL_authentication_mode> ]
            [ -W Wallet_location ]
            [ -P Wallet_password ]
            [ -help | -usage ]
```

The following example deregisters a profile entry and dissociates it from the configuration set 2 (`config 2`) entry:

```
ldapDeleteConn.sh name HRMS config 2
```

Note: Alternatively, you can use the `deleteprofile` option of the Directory Integration and Provisioning Assistant to perform this operation.

The StopOdiServer.sh Tool Syntax

In a client-only installation where OID Monitor and OIDCTL tools are not available, you can start the directory integration and provisioning server without OIDCTL. 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–35 Arguments for Stopping the Oracle Directory Integration and Provisioning Server

Argument	Description
`LDAPhost`	Directory server host. The default is the current host.
`LDAPport`	Directory 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 Oracle directory integration and provisioning server to stop.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.3.2.2-1 or later. Visit:
  `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit:
  `http://www.datafocus.com/`

The schemasync Tool Syntax

The schemasync tool 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: the `-ldap` parameter is optional. 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`

The Oracle Directory Integration and Provisioning Server Registration Tool (odisrvreg)

To register an Oracle directory integration and provisioning server with the directory, this tool creates an entry in the directory and sets the password for the directory integration and provisioning server. 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_`*`hostname`*, at `$`*`ORACLE_HOME`*`/ldap/odi/conf`. This file acts as a private wallet for the directory integration and provisioning server, which uses it on startup to bind to the directory.

Table A–36 describes the parameters that you use with the Oracle Directory Integration and Provisioning Server Registration Tool. You can also run `odisrvreg` in SSL mode to make communication between the tool and the directory fully secure, using the `-U`, `-W`, and `-P` parameters that are also described in Table A–36.

To register the directory integration and provisioning server, enter this command:

```
odisrvreg -h host_name -p port -D binddn -w bindpasswd -I passwd [-U ssl_mode -W
wallet -P wallet_password]
```

Table A–36 Descriptions of ODISRVREG Arguments

Argument	Description
`-h` *`host_name`*	Oracle directory server host name
`-p` *`port_number`*	Port number on which the directory server is running
`-D` *`binddn`*	Bind DN. The bind DN must have authorization to create the registration entry for the directory integration and provisioning server
`-lhost`	In a cold failover cluster configuration, the virtual hostname
`-w` *`bindpasswd`*	Bind password
`-U` *`SSL mode`*	For no authorization, specify `0`. For one-way authorization, specify `1`.
`-W` *`Wallet location`*	Location of the Oracle Wallet containing the SSL certificate
`-P` *`Wallet password`*	Wallet password to open the Oracle wallet

The Provisioning Subscription Tool (oidprovtool) 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 and Provisioning 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 for each application for each subscriber.

Note: To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:

- Cygwin 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

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–37 Provisioning Subscription Tool Parameters

Name	Description	Operations	Mandatory/Optional
`operation`	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 for each invocation of the tool.	all	M
`ldap_host`	Host-name of the directory server on which the subscription operations are to be performed. If not specified, the default value of 'localhost' is assumed.	all	O
profile_status	The status of the profile (`ENABLED`/ `DISABLED`). Default is `ENABLED`.	Create	O
profile_mode	`IBOUND`/`OUTBOUND`/`BOTH`. Default is `OUTBOUND`.	Create	O
profile_debug	The debugging level with which the profile is executed by the Oracle directory integration and provisioning server.	All	O
sslmode	Indicator of whether to execute the Provisioning Subscription Tool in SSL mode. A value of `0` indicates non-ssl and `1` indicates SSL mode.	All	O
`ldap_port`	The TCP/IP port on which the LDAP server is listening for requests. If not specified, the default value of '389' is assumed.	all	O

Table A–37 (Cont.) Provisioning Subscription Tool Parameters

Name	Description	Operations	Mandatory/Optional
`ldap_user_dn`	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.	all	M
`ldap_user_password`	The password of the user on whose behalf the operation is to be performed.	all	M
`application_dn`	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.	all	M
`organization_dn`	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.	all	M
`interface_name`	Database schema name for the PLSQL package. Format of the value should be: [Schema].[PACKAGE_NAME]	create only	M

Table A–37 (Cont.) Provisioning Subscription Tool Parameters

Name	Description	Operations	Mandatory/Optional
`interface_type`	The type of the interface to which events have to be propagated.Valid Values: PLSQL (if not specified this is assumed as the default)	create only	O
`interface_connect_info`	Database connect string Format of this string:[HOST]:[PORT]:[SID]: [USER_ID]:[PASSWORD]	create only	M
`interface_version`	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.	create only	O
`interface_additional_info`	Additional information for the interface. This is not currently used.	create only	O

Table A–37 (Cont.) Provisioning Subscription Tool Parameters

Name	Description	Operations	Mandatory/Optional
`schedule`	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.	create only	O
`max_retries`	The number of times the Provisioning Service should retry a failed event delivery. If not specified, a default value of 5 is assumed.	create only	O
`event_subscription`	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>:DELETEqQthat is, send user and group delete notifications under the organization DN.	create only	O

OID Database Password Utility (oidpasswd) Syntax

The OID Database Password Utility is used to:

- Change the password to the Oracle Internet Directory database.

  Oracle Internet Directory uses a password when connecting to an Oracle database. The default for this password matches the value you specified during installation for the Oracle Application Server administrator's password. You can change this password by using the OID Database Password Utility.

- Create a wallet, named oidlpwdldap1, for the Oracle Internet Directory database password, and a wallet, named `oidpwdr`*`sid`*, for the Oracle directory replication server password.

  The *`sid`* is obtained not from the environment variable *`SID`* but from the connected database.

  With the `create_wallet=true` option, you need to provide the ODS password to authenticate yourself to the ODS database before the ODS wallet can be generated. Note that the default ODS password is the same as that for the Oracle Application Server administrator.

- Unlock a locked directory superuser account, namely, `cn=orcladmin`.

The OID Database Password Utility syntax is:

```
oidpasswd  [connect=connect_string ]  [change_oiddb_pwd=true |
create_wallet=true | unlock_su_acct=true]
```

Changing the Password to the Oracle Internet Directory Database

To change the Oracle Internet Directory database password, enter

```
oidpasswd [connect=connect string ][change_oiddb_pwd=true]
```

If no options are provided, the tool still changes the Oracle Internet Directory database password.

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=`*`connect_string`* option.

For example:

```
$ oidpasswd
current password: ods
new password: newsupersecret
confirm password: newsupersecret
password set.
```

Note:

- User responses are not echoed to the screen when you enter a password.
- Whenever you change the password to the Oracle Internet Directory database by using the OID Database Password Utility, you should also run the `oidemdpasswd` utility. This enables the Oracle Enterprise Manager Daemon (a component of Oracle Enterprise Manager) to properly cache that password and contact the ODS schema upon starting up. Once you have run the `oidemdpasswd` utility, you can monitor Oracle Internet Directory processes from the Oracle Enterprise Manager.

Creating Wallets for the Oracle Internet Directory Database Password and the Oracle Directory Replication Server Password

To create wallets for the Oracle Internet Directory database password and the directory replication server password, enter:

```
oidpasswd [connect=connect string] create_wallet=true
```

The argument `create_wallet` is mandatory in this case. Except for connect string, no other option can be specified.

Unlocking a Super User Account

To unlock a locked account for the directory super user, `cn=orcladmin`, enter:

```
oidpasswd [connect=connect string] unlock_su_acct=true
```

The argument `unlock_su_acct` is mandatory. Except for connect string, no other option can be specified.

OID Database Statistics Collection Tool (oidstats.sh) 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. 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 Oracle Internet Directory daemons.

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 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

The OID Database Statistics Collection Tool uses this syntax:

```
oidstats.sh [ -connect connect_string ]
            [ -all ]
            [ -cat catalog_name ]
            [ -pct percent ]
           [ -help | -usage ]
```

The parameters are:

Parameter	Description	Default
`connect` *connect_string*	Database 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
```

The OID Migration Tool (ldifmigrator) Syntax

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 [options] {parameter_name=value ...}
{s_SubVar=value ... }
```

Table A–38 describes the command-line parameters used by this tool in further detail:

Table A–38 ldifmigrator Parameters

Parameter	Mandatory?	Description
Input_file	Yes	The file containing the substitution variables

Table A–38 (Cont.) ldifmigrator Parameters

Parameter	Mandatory?	Description
`Output_file`	Yes	The name of the file to be generated by this tool
`-lookup`	No	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 specified using host parameters. The host is mandatory when `-lookup` flag is specified.
`Host`	Yes (only in lookup mode)	The directory server name. This parameter is mandatory when `-lookup` flag is specified.
`Port`	No	The port on which the directory server is listening. If not specified the port `389` will be used
`DN`	Yes (only in lookup mode)	Bind DN. This is a mandatory parameter when `-lookup` flag is specified.
`Password`	No	Bind password
`Subscriber`	No	The subscriber whose attributes will be used as substitution variable. If not specified, then the default identity management realm specified in the Root Oracle Context will be used.
`s_SubsVar1.N`	No	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 in the Oracle Internet Directory.

Table A–39 Predefined 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 realm- specific Oracle context.
%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 realm- specific Oracle context.
%s_UserNicknameAttribute%	The nickname attribute to be used for user entries in the identity management realm.	This is assigned the value of the attribute: `orclCommonNicknameAttribute` from the entry `cn=Common,cn=Products` under the realm- specific Oracle context.
%s_SubscriberDN%	Distinguished name of the LDAP entry corresponding to the identity management realm.	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 realm-specific Oracle Context.	First the realm DN is computed as described earlier 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`.
%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 previous 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 following table 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 previous 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 this 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 these variables, the OID Migration Tool also honors the command-line variable called `s_UserOrganization` and substitutes all occurrences of it with the value `Development`. In this case the output of the tool stored in `sample.ldif` is 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 this 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:

Table A–40 Substitution Variables for the subscriber "acme"

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 these 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
```

Load Capability

Using the load capability the users of this tool could directly load the data into Oracle Internet Directory. If an entry is already present in the directory then that directory entry will be logged to the file. The addition of the directory entries could fail for other reasons as well, for instance not enough permission to add or parent entry not being present. The command line tool will now take a new option `-load`, which will load the user information to the directory.

Reconcile Capability

The user migration tool capabilities available in Oracle Application Server 10*g* (9.0.4) are only useful when an older version of the *i*AS component is the only source of truth for all users being migrated to Oracle Internet Directory. However, in a practical deployment, the following scenarios arise:

- The users to be migrated have already been defined in Oracle Internet Directory.

- More than one distinct application needs to be migrated to Oracle Internet Directory.

To address these requirements, a new option `-reconcile`, has been added to the user migration tool. This option requires an argument: `-reconcile SAFE | SAFE_EXTENDED | NORMAL`.

Table A–41 Different Modes for Use of -reconcile

Optional Arguments	Description
-reconcile SAFE	checks for the existence of the user entry in the directory
–reconcile NORMAL	checks that all the new attributes will be added and those attributes that are already present in the Oracle Internet Directory will have their values replaced with the new ones
–reconcile SAFE_ EXTENED	all the new attributes will be added, but for the existing attributes if you try to add a new value then it will add this new value to the existing set of values.

Example 47–1 -reconcile SAFE option.

This option should be used when the user would like to append the only those attributes that are not already present in the directory. In the case of the above user entry, the user migration tool will parse this LDIF entry and substitute the values for `s_subscriber_user_base` and `s_nickname_attr`. After this, the tool will retrieve the jsmith entry from the directory. If the directory does not contain an entry for `jsmith` then it would simply add this entry for the first time. On the other hand if the entry already exists with attributes as defined above then it will add only those attributes that are not present in directory. In the above case it will add only `homePhone` and `homePostalAddress`.

Now the `Jsmith` entry in the directory will be:

```
dn: cn=jsmith, dc=oracle, dc=com
cn: jsmith
sn: Smith
orclGlobalID: 86A8485163303EBEE034080020AB67AA
uid: jsmith
objectClass: inetOrgPerson
objectClass: orclUser2
title: Member of Technical Staff
homePhone: 650-584-5670
homePostalAddress: 232 Gonzalez Drive$ San Francisco$ CA$ 94404
```

Example 47–2 -reconcile NORMAL option.

This option can be used when the user would like to overwrite attributes that are present in the directory. In the case of the above user entry, the user migration tool will parse this LDIF entry and substitute the values for `s_subscriber_user_base` and `s_nickname_attr`. After this, the tool will retrieve the `jsmith` entry from the directory. If the directory does not contain an entry for `jsmith` then it would simply add this entry for the first time. On the other hand if the entry already exists with attributes as defined above then it will add only those attributes that are not present in directory. In addition to this the attribute that is already present will be deleted and freshly added with new value. In the above case it will add `homePhone` and `homePostalAddress` and replace the attribute value for the attribute `title` with the new value.

Now the `Jsmith` entry in the directory will be:

```
dn: cn=jsmith, dc=oracle, dc=com
cn: jsmith
sn: Smith
orclGlobalID: 86A8485163303EBEE034080020AB67AA
uid: jsmith
objectClass: inetOrgPerson
objectClass: orclUser2
title: Principle Member of Technical Staff
homePhone: 650-584-5670
homePostalAddress: 232 Gonzalez Drive$ San Francisco$ CA$ 94404
```

Example 47–3 -reconcile SAFE_EXTENDED option.

This option can be used when the user would like to add the values to existing attributes. In the case of the above user entry, the user migration tool will parse this LDIF entry and substitute the values for `s_subscriber_user_base` and `s_nickname_attr`. After this, the tool will retrieve the `jsmith` entry from the directory. If the directory does not contain an entry for `jsmith` then it would simply add this entry for the first time. On the other hand if the entry already exists with attributes as defined above then it will add the attributes `homePhone` and `homePostalAddress` and the new value will be added to the existing `title` attribute.

Now the `Jsmith` entry in the directory will be:

```
dn: cn=jsmith, dc=oracle, dc=com
cn: jsmith
sn: Smith
orclGlobalID: 86A8485163303EBEE034080020AB67AA
uid: jsmith
objectClass: inetOrgPerson
objectClass: orclUser2
```

```
title: Member of Technical Staff
title: Principle Member of Technical Staff
homePhone: 650-584-5670
homePostalAddress: 232 Gonzalez Drive$ San Francisco$ CA$ 94404
```

Table A–42 -reconcile SAFE type LDIF records

Sno	Entry Changetype	Attribute Changetype	Action
1	Add/No Change type	-	Add only new attributes.
2	Modrdn/Moddn	-	The ldifmigrator tool will not support this change type.
3	Delete	-	Do not delete the entry from the directory.
4	Modify	add	Add this attribute. If the entry doesn't exist in the directory then ignore the record as invalid. If the attribute does not exist then add this attribute, otherwise ignore.
5	-do-	replace	If the entry does not contain the attribute then it will be added. Otherwise Ignore change to the attribute, that is, do not apply the change. When the entry is not present in the directory then ignore it as the invalid entry.
6	-do-	delete	Ignore the change to the attribute, that is, do not apply the change.

Table A–43 -reconcile NORMAL type LDIF records

Sno	Entry Changetype	Attribute Changetype	Description
1	Add/No Change type	-	Adds the attributes that are not populated in the directory and replaces the attributes that are already populated
2	Modrdn/Moddn	-	The ldifmigrator tool will not support this change type.
3	Delete	-	Delete the entry from the directory

Table A–43 (Cont.) -reconcile NORMAL type LDIF records

Sno	Entry Changetype	Attribute Changetype	Description
4	Modify	add	If entry doesn't contain the attribute then it will be added. When it contains the attribute then replace it with the specified attribute. If the entry doesn't exist in the directory then ignore the record as invalid.
5	-do-	replace	If entry doesn't contain the attribute then it will be added. When it contains the attribute then replace it with the specified attribute. If the entry itself does not exist in the directory then ignore the record as invalid
6	-do-	delete	Remove the specified attribute from the directory.

Table A–44 -reconcile SAFE_EXTENDED type LDIF records

Sno	Entry Changetype	Attribute Changetype	Description
1	Add/No Change type	-	Add only new attributes. If the entry does not exist then create a new entry.
2	Modrdn/Moddn	-	The ldifmigrator tool will not support this change type.
3	Delete	-	Do not delete the entry from the directory.
4	Modify	add	Add this attribute. If the entry doesn't exist in the directory then ignore the record as invalid. If the attribute does not exist then add this attribute, otherwise add the new values to the directory.
5	-do-	replace	If the entry does not contain the attribute then it will be added. Otherwise Ignore change to the attribute, that is, do not apply the change. When the entry is not present in the directory then ignore it as the invalid entry.
6	-do-	delete	Ignore the change to the attribute, that is, do not apply the change.

OID Migration Tool Error Messages

The OID Migration Tool can display these error messages:

Table A–45 Error Messages of OID Migration Tool

Message	Reason	Remedial Action
Environment variable *ORACLE_HOME* not defined	*ORACLE_HOME* is not defined.	Set the environment variable *ORACLE_HOME*
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

Table A–45 (Cont.) Error Messages of OID Migration Tool

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 identity management realm does not exist in the directory	Check the realm parameter
Not all the substitution variables are defined in the directory server specified	If the identity management realm entry does not contain the required attributes, then this error occurs.	Check the realm entry in the directory
Error occurred while migrating LDIF data to Oracle Internet Directory	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.

B

Oracle Internet Directory 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, the ldapext charter and LDAP drafts, and 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), each of which is available on the IETF Web site at: `http://www.ietf.org`.

Table B–1 RFCs Enforced by Oracle Internet Directory

RFC	Title
1777	Lightweight Directory Access Protocol
1778	The String Representation of Standard Attribute Syntaxes
1779	A String Representation of Distinguished Names
1960	A String Representation of LDAP Search Filters
2079	Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs)
2247	Using Domains in LDAP/X.500 Distinguished Names
2251	Lightweight Directory Access Protocol (v3)
2252	Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions
2253	Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names
2254	The String Representation of LDAP Search Filters
2255	The LDAP URL Format
2256	A Summary of the X.500(96) User Schema for use with LDAPv3

IETF Drafts Enforced by Oracle Internet Directory

Oracle Internet Directory enforces the following two drafts of the IETF, each of which is available on the IETF Web site at: `http://www.ietf.org`.

- "Definition of the inetOrgPerson LDAP Object Class"
- "Referrals and Knowledge References in LDAP Directories"

Proprietary Schema Elements of Oracle Internet Directory

Oracle Internet Directory's proprietary schema includes attributes and object classes in these categories:

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 Schema Elements

Table B–2 Access Control Schema Elements

Object Class	Attributes
orclPrivilegeGroup	orclEntryLevelACI, orclACI

Audit Log Schema Elements

Table B–3 Audit Log Schema Elements

Object Class	Attributes
OrclAuditOC	orclServerEvent, orcleventtype, orclauditattribute, orclauditmessage, orcleventtime, orcluserdn, orclSequence, orclAuditLevel, orclOpResult

Attribute Uniqueness Schema Elements

Table B–4 Attribute Uniqueness Constraint Entry

Attribute Name	Mandatory?	Valid Value	Default Value	Default Effect
orcluniqueattrname	Yes	Any string	N/A	N/A
orcluniquescope	No	One of the following: ■ base—Searches the root entry only ■ onelevel—Searches one level only ■ sub—Searches the entire directory	sub	Searches the entire directory

Table B–4 (Cont.) Attribute Uniqueness Constraint Entry

Attribute Name	Mandatory?	Valid Value	Default Value	Default Effect
orcluniqueenable	No	Either 0 (disable) or 1 (enable)	0	Disables attribute uniqueness
orcluniquesubtree	No	Any string	" "	Searches the entire directory
orcluniqueobjectclass	No	Any string	" "	Searches all object classes

See Also: "Enabling and Disabling Attribute Uniqueness by Using Command-Line Tools" on page 8-9

Configuration Set Entry Schema Elements

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.

Table B–5 Configuration Set Entry Attributes

Attribute	Description
orcldebugflag	Debug level associated with this instance of the server. The default for configset0 is 0. The range is 0 to 67108863.
orclmaxcc	Maximum number of concurrent database connections. The default for configset0 is 10. You cannot use a negative value for this attribute.
orclserverprocs	Number of server processes to start. The default for configset0 is 1. You cannot use a negative value for this attribute.
orclsslport	SSL mode default port (default 636). When you run the directory in the secure mode, it listens at default port 636 and accepts only SSL-based TCP/IP connections. (When you run the directory in the normal mode, it listens at default port 389, accepting normal TCP/IP connections.) You might want to change this port when you add multiple LDAP server instances.
orclnonsslport	Non-SSL mode default port (default 389).

Table B–5 (Cont.) Configuration Set Entry Attributes

Attribute	Description
orclsslenable	Flag for enabling or disabling SSL. You would want to use this flag when you use different instances of the same server for either SSL or non-SSL. You may use one of the following values: ■ 0—for non-secure operation only ■ 1—for SSL authentication only ■ 2— for both non-secure operation and SSL authentication 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 = 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. ■ 32 = One-way authentication. Only the directory server authenticates itself to the client by sending its certificate to the client. ■ 64 = Two-way authentication. Both 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 UNIX, 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`

orclsslversion	SSL version. The default is 3.

Debug Logging Schema Elements

Table B–6 Debug Logging Schema Elements

Attribute	Description
orcldebugforceflush	Specifies whether debug messages are to be written to the log file when a message is logged by the directory server. To enable it, set its value to 1. To disable it set it to 0, which is its default value. **See Also:** "Force Flushing the Trace Information to a Log File" on page 10-9
orcldebugop	To make logging more focused, limits logged information to particular directory server operations by specifying the debug dimension to those operations. **See Also:** "Setting the Operation Debug Dimension" on page 10-8

Dynamic Groups Schema Elements

Table B–7 lists and describes the attributes of the orclDynamicGroup object class

Table B–7 orclDynamicGroup Attributes for "Connect By" Assertions

Attribute	Description
orclConnectByAttribute	The attribute that you want to use as the filter for the query—for example, manager
orclConnectByStartingValue	The DN of the attribute you specified in the orclConnectByAttribute attribute—for example, Anne Smith

See Also: "Dynamic Groups" on page 9-3 for information about dynamic groups and "connect by" assertions

Garbage Collection Schema Elements

Table B–8 Garbage Collection Configuration Parameters

Attribute	Description	Mandatory?	Default Value
`orclPurgeBase`	The base DN of DIT where the garbage collection task is applied	Yes	RDN of garbage collector configuration entry DN
`orclpurgestart`	Time in seconds when the garbage collector starts to run. If the garbage collector is enabled, and the value for this attribute is `0`, then the garbage collector is enabled immediately. The format is `yyyymmddhhmmss`.	No	NULL
`orclpurgetargetage`	Age of the target objects in hours. All objects older than the age specified by this attribute are purged.	No	12 (or 10 days old if the attribute value not specified)
`orclPurgeInterval`	Time interval in hours that the garbage collection job is executed again. This can be measured from either the point in time specified in the `orclpurgestart` attribute or from the last time it was run	No	24
`orclpurgetransize`	Number of objects to be purged in one commit transaction.	No	1000
`orclpurgerun`	Indicator that the submitted job is to be executed immediately whenever this attribute is added or modified	No	N/A
`orclPurgeEnable`	Flag to enable or disable garbage collectors	No	1
`orclPurgeDebug`	Flag to enable or disable collection of debugging messages	No	0
`orclpurgefilename`	Name of file that stores garbage collection logging messages	No	oidgc001
`orclpurgefileloc`	Absolute file directory where the log file is saved	No	. (period)

Schema Elements for Predefined Garbage Collectors

Oracle Internet Directory provides several predefined garbage collectors that, together, clean up all unwanted data in the directory server. These predefined garbage collectors are:

- Audit Log Garbage Collector

- Change Log Garbage Collector
- General Statistics Garbage Collector
- Health Statistics Garbage Collector
- Security and Refresh Events Garbage Collector
- System Resource Events Garbage Collector
- Tombstone Garbage Collector

Audit Log Garbage Collector

Audit log garbage collector cleans up unwanted entries created for auditing the directory server.

Table B–9 Attributes for the Audit Log Garbage Collector

Attribute	Description	Default Value
`orclPurgeBase`	The base DN of the naming context to which the garbage collection task is to be applied.	`cn=auditlog`
`orclpurgestart`	Time in seconds when the garbage collector starts to run. If the garbage collector is enabled, and the value for this attribute is `0`, then the garbage collector is enabled immediately. The format is `yyyymmddhhmmss`.	NULL (12:00 a.m. of the day Oracle Internet Directory is installed)
`orclpurgetargetage`	The age of the target objects in hours. All the objects older than the age specified by this attribute are purged.	12 hours
`orclPurgeInterval`	Time interval in hours that the garbage collection job is executed again. This can be measured from either the point in time specified in the `orclpurgestart` attribute or from the last time it was run	NULL (24 hours)
`orclpurgetransize`	The number of objects to be purged in one commit transaction.	1000
`orclpurgerun`	Every time this attribute is added or modified, then the submitted job is executed immediately.	N/A
`orclPurgeEnable`	Flag to enable/disable garbage collectors	1
`orclPurgeDebug`	Flag to enable/disable debugging messages collecting	0
`orclpurgefilename`	File name that saves garbage collection logging messages	`oidgc001.log`
`orclpurgefileloc`	Absolute file directory where the log file is saved.	. (period)

Change Log Garbage Collector

Change log garbage collector cleans up the consumed change log entries in the directory.

Table B–10 Attributes of the Change Log Garbage Collector

Attribute	Description	Default Value
orclPurgeBase	The base DN of the naming context to which the garbage collection task is to be applied.	cn=changelog
orclpurgestart	Time in seconds when the garbage collector starts to run. If the garbage collector is enabled, and the value for this attribute is 0, then the garbage collector is enabled immediately. The format is yyyymmddhhmmss.	NULL (12:00 a.m. of the day Oracle Internet Directory is installed)
orclpurgetargetage	The age of the target objects in hours. All the objects older than the age specified by this attribute are purged.	12 hours
orclPurgeInterval	Time interval in hours that the garbage collection job is executed again. This can be measured from either the point in time specified in the orclpurgestart attribute or from the last time it was run	NULL (24 hours)
orclpurgetransize	The number of objects to be purged in one commit transaction.	1000
orclpurgerun	Every time this attribute is added or modified, then the submitted job is executed immediately.	N/A
orclPurgeEnable	Flag to enable/disable garbage collectors	1
orclPurgeDebug	Flag to enable/disable debugging messages collecting	0
orclpurgefilename	File name that saves garbage collection logging messages	oidgc001.log
orclpurgefileloc	Absolute file directory where the log file is saved.	. (period)

General Statistics Garbage Collector

The General Statistics garbage collector cleans up unwanted general statistical entries created for the directory server.

Table B–11 Attributes of the General Statistics Garbage Collector

Attribute	Description	Default Value
orclPurgeBase	The base DN of the naming context to which the garbage collection task is to be applied.	cn=orclgeneralstats,cn=orclsm
orclpurgestart	Time in seconds when the garbage collector starts to run. If the garbage collector is enabled, and the value for this attribute is 0, then the garbage collector is enabled immediately. The format is *yyyymmddhhmmss*.	NULL (12:00 a.m. of the day Oracle Internet Directory is installed)
orclpurgetargetage	The age of the target objects in hours. All the objects older than the age specified by this attribute are purged.	12 hours
orclPurgeInterval	Time interval in hours that the garbage collection job is executed again. This can be measured from either the point in time specified in the orclpurgestart attribute or from the last time it was run	NULL (24 hours)
orclpurgetransize	The number of objects to be purged in one commit transaction.	1000
orclpurgerun	Every time this attribute is added or modified, then the submitted job is executed immediately.	N/A
orclPurgeEnable	Flag to enable/disable garbage collectors	1
orclPurgeDebug	Flag to enable/disable debugging messages collecting	0
orclpurgefilename	File name that saves garbage collection logging messages	oidgc001.log
orclpurgefileloc	Absolute file directory where the log file is saved.	. (period)

Health Statistics Garbage Collector

The Health Statistics garbage collector cleans up unwanted health statistics entries created for the directory server.

Table B–12 Attributes of the Health Statistics Garbage Collector

Attribute	Description	Default Value
orclPurgeBase	The base DN of the naming context to which the garbage collection task is to be applied.	cn=orclhealthstats, cn=orclsm
orclpurgestart	Time in seconds when the garbage collector starts to run. If the garbage collector is enabled, and the value for this attribute is 0, then the garbage collector is enabled immediately. The format is *yyyymmddhhmmss*.	NULL (12:00 a.m. of the day Oracle Internet Directory is installed)
orclpurgetargetage	The age of the target objects in hours. All the objects older than the age specified by this attribute are purged.	12 hours
orclPurgeInterval	Time interval in hours that the garbage collection job is executed again. This can be measured from either the point in time specified in the orclpurgestart attribute or from the last time it was run.	NULL (24 hours)
orclpurgetransize	The number of objects to be purged in one commit transaction.	1000
orclpurgerun	Every time this attribute is added or modified, then the submitted job is executed immediately.	N/A
orclPurgeEnable	Flag to enable/disable garbage collectors	1
orclPurgeDebug	Flag to enable/disable debugging messages collecting	0
orclpurgefilename	File name that saves garbage collection logging messages	oidgc001.log
orclpurgefileloc	Absolute file directory where the log file is saved.	. (period)

Security and Refresh Events Garbage Collector

The Security and Refresh Events garbage collector cleans up the unwanted entries created for monitoring the security and refresh events of the directory server.

Table B–13 Attributes of the Security and Refresh Events Garbage Collector

Attribute	Description	Default Value
orclPurgeBase	The base DN of the naming context to which the garbage collection task is to be applied.	cn=orclsecrefreshevents,cn=orclsm
orclpurgestart	Time in seconds when the garbage collector starts to run. If the garbage collector is enabled, and the value for this attribute is 0, then the garbage collector is enabled immediately. The format is *yyyymmddhhmmss*.	NULL (12:00 a.m. of the day Oracle Internet Directory is installed)
orclpurgetargetage	The age of the target objects in hours. All the objects older than the age specified by this attribute are purged.	12 hours
orclPurgeInterval	Time interval in hours that the garbage collection job is executed again. This can be measured from either the point in time specified in the orclpurgestart attribute or from the last time it was run.	NULL (24 hours)
orclpurgetransize	The number of objects to be purged in one commit transaction.	1000
orclpurgerun	Every time this attribute is added or modified, then the submitted job is executed immediately.	N/A
orclPurgeEnable	Flag to enable/disable garbage collectors	1
orclPurgeDebug	Flag to enable/disable debugging messages collecting	0
orclpurgefilename	File name that saves garbage collection logging messages	oidgc001.log
orclpurgefileloc	Absolute file directory where the log file is saved.	. (period)

System Resource Events Garbage Collector

The System Resource Events garbage collector cleans up unwanted entries created for monitoring system resources events of the directory server.

Table B–14 Attributes of the System Resource Events Garbage Collector

Attribute	Description	Default Value
orclPurgeBase	The base DN of the naming context to which the garbage collection task is to be applied.	cn=orclsysresourceevents, cn=orclsm
orclpurgestart	Time in seconds when the garbage collector starts to run. If the garbage collector is enabled, and the value for this attribute is 0, then the garbage collector is enabled immediately. The format is *yyyymmddhhmmss*.	NULL (12:00 a.m. of the day Oracle Internet Directory is installed)
orclpurgetargetage	The age of the target objects in hours. All the objects older than the age specified by this attribute are purged.	12 hours
orclPurgeInterval	Time interval in hours that the garbage collection job is executed again. This can be measured from either the point in time specified in the orclpurgestart attribute or from the last time it was run.	NULL (24 hours)
orclpurgetransize	The number of objects to be purged in one commit transaction.	1000
orclpurgerun	Every time this attribute is added or modified, then the submitted job is executed immediately.	N/A
orclPurgeEnable	Flag to enable/disable garbage collectors	1
orclPurgeDebug	Flag to enable/disable debugging messages collecting	0
orclpurgefilename	File name that saves garbage collection logging messages	oidgc001.log
orclpurgefileloc	Absolute file directory where the log file is saved.	. (period)

Tombstone Garbage Collector

The Tombstone garbage collector cleans up unwanted entries marked as deleted.

Table B–15 Attributes of the Tombstone Garbage Collector

Attribute	Description	Default Value
`orclPurgeBase`	The base DN of the naming context to which the garbage collection task is to be applied.	`cn=tombstone`
`orclpurgestart`	Time in seconds when the garbage collector starts to run. If the garbage collector is enabled, and the value for this attribute is `0`, then the garbage collector is enabled immediately. The format is *yyyymmddhhmmss*.	NULL (12:00 a.m. of the day Oracle Internet Directory is installed)
`orclpurgetargetage`	The age of the target objects in hours. All the objects older than the age specified by this attribute are purged.	12 hours
`orclPurgeInterval`	Time interval in hours that the garbage collection job is executed again. This can be measured from either the point in time specified in the `orclpurgestart` attribute or from the last time it was run.	NULL (24 hours)
`orclpurgetransize`	The number of objects to be purged in one commit transaction.	1000
`orclpurgerun`	Every time this attribute is added or modified, then the submitted job is executed immediately.	N/A
`orclPurgeEnable`	Flag to enable/disable garbage collectors	1
`orclPurgeDebug`	Flag to enable/disable debugging messages collecting	0
`orclpurgefilename`	File name that saves garbage collection logging messages	oidgc001.log
`orclpurgefileloc`	Absolute file directory where the log file is saved.	. (period)

Oracle Internet Directory Plug-In for Garbage Collection

The garbage collection framework relies on the Oracle Internet Directory plug-in framework to trigger the garbage collection engine. This section tells you the attribute value pairs that the garbage collection plug-in uses for various operations.

Attributes for Creating a Garbage Collector

To create a garbage collector, the garbage collection plug-in uses the attribute value pairs listed in Table B–16.

Table B–16 Attribute Value Pairs for Creating a Garbage Collector

Attribute	Value
orclpluginname	PurgeAdmin
orclplugintype	operational
orclplugintiming	post
orclpluginldapoperation	ldapadd
orclpluginsubscriberdnlist	cn=purgeconfig,cn=subconfigsubentry

Attributes for Modifying a Garbage Collector

To modify a garbage collector, the garbage collection plug-in uses the attribute value pairs listed in Table B–17.

Table B–17 Attribute Value Pairs for Modifying a Garbage Collector

Attribute	Value
orclpluginname	PurgeAdmin
orclplugintype	operational
orclplugintiming	post
orclpluginldapoperation	ldapmodify
orclpluginsubscriberdnlist	cn=purgeconfig,cn=subconfigsubentry

Attributes for Deleting a Garbage Collector

To delete a garbage collector, the garbage collection plug-in uses the attribute value pairs listed in Table B–18.

Table B–18 Attribute Value Pairs for Deleting a Garbage Collector

Attribute	Value
orclpluginname	PurgeAdmin
orclplugintype	operational
orclplugintiming	post
orclpluginldapoperation	ldapdelete
orclpluginsubscriberdnlist	cn=purgeconfig,cn=subconfigsubentry

Optional Attributes of the orclUserV2 Object Class

The following are optional attributes from the orclUserV2 object class:

Table B–19 Attributes in the orclUserV2 Object Class

Attribute	Description
OrclPassword	Identifies an Oracle-specific password for custom authentication schemes like O3Logon for the database server
OrclHireDate	Specifies the date on which an employee starts working for a company
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

Table B–19 (Cont.) Attributes in the orclUserV2 Object Class

Attribute	Description
`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`	Specifies the date on which the user can successfully begin to authenticate to the Oracle Application Server Single Sign-On server. Values are represented in Universal Time format.
`orclActiveEnddate`	Specifies the date after which the user can no longer authenticate to the Oracle Application Server Single Sign-On server. Values are represented in Universal Time format.

Oracle Directory Integration and Provisioning Platform Schema Elements

Table B–20 Attributes in Integration Profiles for Third-Party Directories

Attribute	Description
General Information	-
Profile Name (`orclodipAgentName`)	Name of the profile for the particular third-party directory you are integrating with. This attribute is mandatory.
Synchronization Mode (`orclodipSynchronizationMode`)	Direction of synchronization between Oracle Internet Directory and the connected directory. `IMPORT` indicates importing changes from the third-party directory to Oracle Internet Directory. `EXPORT` indicates exporting changes from Oracle Internet Directory to the third-party directory.
ProfileStatus (`orclOdipAgentControl`)	Indicator whether the profile is enabled or disabled. The default is DISABLE. You must set this value to ENABLE.
Profile Password (`orclodipProfilePassword`)	The password used by the profile to bind to Oracle Internet Directory. In case of import, the changes are made with the profile name as the identity. The default value is `welcome`. **Note:** For security reasons, change this password.

Table B–20 (Cont.) Attributes in Integration Profiles for Third-Party Directories

Attribute	Description
Scheduling Interval (`orclODIPSchedulingInterval`)	Time interval in seconds after which a connected directory is synchronized with Oracle Internet Directory. The default is `600`. This attribute can be modified.
Maximum Number of Retries (`orclodipSyncRetryCount`)	Maximum number of times Oracle directory integration and provisioning server tries to run the third-party directory connector in the event of a failure. The default is `5`.
Profile Version	Version of the Oracle Directory Integration and Provisioning platform with which this profile was created.The default value is `1.0`. This value cannot be modified.
Debug Level (`orclodipdebuglevel`)	Identifier indicating the level of debugging required for any profile. Set this attribute to 63 for the maximum debug level. **See Also:** "Setting Debug Logging Levels" on page 10-6
Execution Information	-
Agent Execution Command (`orclodipAgentExeCommand`)	Connector executable name and argument list used by the directory integration and provisioning server. It can be passed as a command-line argument when the connector is invoked. **See Also:** Chapter 39, "Synchronization with Oracle Human Resources" for typical usage of passing it in the command-line
Connected Directory Account (`orclodipConDirAccessAccount`)	Valid user account in the connected directory to be used by the connector for synchronization. The value is specific to the connected directory with which you are integrating. For instance, for the SunONE synchronization connector, it is the valid bind DN in the SunONE Directory Server. For the Human Resources Connector, it is a valid user identifier in the Oracle Human Resources database. For other connectors, it can be passed as a command-line argument when the connector is invoked. **See Also:** Chapter 39, "Synchronization with Oracle Human Resources" for typical usage of passing it in the command-line

Table B–20 (Cont.) Attributes in Integration Profiles for Third-Party Directories

Attribute	Description
Connected Directory Account Password (`orclodipConDirAccessPassword`)	Password to be used by the user specified in the `orclOdipConDirAccessAccount` attribute to connect to the connected directory. The value is specific to the third-party directory with which you are integrating. For instance, for the SunONE synchronization connector, it is the valid bind password in the SunONE Directory Server. For the Human Resources Agent, it is the Oracle Human Resources database password.
Additional Config Info (`orclodipAgentConfigInfo`)	Any configuration information that you want the connector to store in Oracle Internet Directory. It is passed by the directory integration and provisioning server to the connector at time of connector invocation. The information is stored as an attribute and the directory integration and provisioning server does not have any knowledge of its content. When the connector is scheduled for execution, the value of the attribute is stored in the file, *`$ORACLE_HOME`*`/ldap/odi/conf/`*`profile_name`*`.cfg` that can be processed by the connector. Upload the file by using either the Directory Integration and Provisioning Assistant or the `ldapuploadagentfile.sh` tool. Do this for both import and export agents. **See Also:** ■ "The Directory Integration and Provisioning Assistant" on page A-107 ■ "The ldapUploadAgentFile.sh Tool Syntax" on page A-120
Connected Directory URL (`orclOdipConDirURL`)	Connect details required to connect to the connected directory. This parameter refers to the host name and port number as *`host`*`:`*`port`*`:`*`sslmode`*. To connect by using SSL, enter *`host`*`:`*`port`*`:1`. Make sure the certificate to connect to the directory is stored in the wallet, the location of which is specified in the file `odi.properties`. **Note:** To connect to SunONE Directory Server by using SSL, the server certificate needs to be loaded into the wallet. **See Also:** The chapter on Oracle Wallet Manager in *Oracle Advanced Security Administrator's Guide*

Table B–20 (Cont.) Attributes in Integration Profiles for Third-Party Directories

Attribute	Description
Interface Type (`orclodipInterfaceType`)	The data format or protocol used in synchronization. Supported values are: ■ LDIF—Import or export from a LDIF File ■ Tagged—Import or export from a tagged file—a proprietary format supported by the Oracle directory integration and provisioning server, similar to LDIF format ■ LDAP—Import from or export to an LDAP-compliant directory ■ DB —Import from or export to an Oracle9*i* Database Server directory
Mapping Information	-
Mapping Rules (`orclodipAttributeMappingRules`)	Attribute for storing the mapping rules. Store the mapping rules in a file by using the Directory Integration and Provisioning Assistant or the `ldapuploadagentfile.sh` tool. **See Also:** ■ "Mapping Rules and Formats" on page 33-5 ■ "Format of the Mapping Rules Attribute" on page 33-7 ■ "The Directory Integration and Provisioning Assistant" on page A-107
Connected Directory Matching Filter (`orclodipConDirMatchingFilter`)	This attribute specifies the filter to apply to the third-party directory change log. It is used in the import profile. The filter must be set in the import profile when both the import and export integration profiles are enabled, as follows: `Modifiersname != connected_directory_account` This prevents the same change from being exchanged between the two directories indefinitely. To avoid confusion, make this account specific to synchronization.

Table B–20 (Cont.) Attributes in Integration Profiles for Third-Party Directories

Attribute	Description
OID Matching Filter (`orclOdipOIDMatchingFilter`)	In export profiles, this attribute specifies the filter to apply to the Oracle Internet Directory change log container. It is used in the export profile. It must be set in the export profile when both the import and export integration profiles are enabled, as in the following example: `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. In import profiles, this attribute specifies a key for mapping entries between Oracle Internet Directory and the connected directory. This is useful when the DN cannot be used as the key.
Status Information	-
OID Last Applied Change Number (`orclLastAppliedChangeNumber`)	For export operations, the last change from Oracle Internet Directory that was applied to the connected directory. The default value is `0`. Set this to the value of the `lastchangenumber` attribute of Oracle Internet Directory. If you have used the Directory Integration and Provisioning Assistant for bootstrapping using LDAP, then this is set automatically at the end of the bootstrapping process. This is valid only in the export profile.
Last Execution Time (`orclodipLastExecutionTime`)	Status attribute set to the last time the integration profile was executed successfully by the Oracle directory integration and provisioning server. Its format is `dd-mon-yyyy hh:mm:ss`, where `hh` is the time of day in 24-hour format. This attribute is initialized during profile creation.
Last Successful Execution Time (`orclodipLastSuccessfulExecution Time`)	Status attribute set to the last time the integration profile was executed successfully by the Oracle directory integration and provisioning server. The format is `dd-mon-yyyy hh:mm:ss`, where `hh` is the hour in 24-hour format.

Table B–20 (Cont.) Attributes in Integration Profiles for Third-Party Directories

Attribute	Description
Synchronization Status	Synchronization status of the last execution: Success or failure. (`orclodipSynchronizationStatus`) Initially, this attribute has the value `Yet to be executed`. It is a read-only attribute
Synchronization Errors (`orclodipSynchronizationErrors`)	Messages explaining errors if the last execution failed. This parameter is updated by Oracle directory integration and provisioning server. It is a read-only attribute.
Last Applied Change Number (`orclodipConDirLastAppliedChgNum`)	For import operations, the last change from the connected directory that was applied to Oracle Internet Directory. The default value is `0`. Set this to the value of the `lastchangenumber` attribute of Oracle Internet Directory. If you have used the Directory Integration and Provisioning Assistant for bootstrapping using LDAP, then this is set automatically at then end of the bootstrapping process. This is valid only in the import profile.

See Also:

- "Updating the Default Parameters" on page 42-7 for instructions specific to integration with SunONE Directory Server
- "Updating the Default Parameters" on page 43-11 for instructions specific to integration with a Microsoft Active Directory with a single domain

Oracle Internet Directory Configuration Schema Elements

Table B–21 Oracle Internet Directory Configuration Parameters

Object Classes	Attributes
subconfig, orclConfigSet, orclLDAPSubConfig, orclREPLSubConfig, orclcontainerOC, subregistry, orclLDAPInstance, orclREPLInstance, orclIndexOC, orcleventLog, orclEvents	orcldebugflag, orclMaxCC, orclDBType, orclSuffix, orclDITRoot, orclSuName, orclSuPassword, orclSizeLimit, orclTimeLimit, orclGuName, orclGuPassword, orclServerProcs, orclconfigsetnumber, orclhostname, orclIndexedAttribute, orclCatalogEntryDN, orclServerMode, orclPrName, orclPrPassword, orclUseEncrypt, orclDirectoryVersion

Oracle Internet Directory Server Manageability Schema Elements

Table B–22 Attributes for Oracle Internet Directory Server Manageability

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. If OrclStatsLevel is enabled—that is, user statistics are turned on—and there are few users, then provide a greater value for this attribute. Conversely, if there are many users, then provide a lesser value.
OrclEventLevel	Specify critical events related to security and system resources that you want recorded. The default is 0—that is, no critical events are recorded. For events other than super user, proxy user, and replication login, set the value of the orclStatsFlag attribute 1. **See Also:** "Configuring Critical Events" on page 10-22 for a list of critical events that can be monitored

Table B–22 (Cont.) Attributes for Oracle Internet Directory Server Manageability

Attribute	Description
`OrclStatsLevel`	Specify the level of statistics collection for users. There is only one valid value in this release, namely, `1`. Specifying this value collects the number of bind and compare operations against the directory and the user who performed each one.
`OrclMaxTcpIdleConnTime`	Specifies maximum TCP connection time in minutes for an idle connection to be recorded as idle. Its default value is 120 minutes (2 hours). Please note that the value of this attribute should be less than that of the DSA Configuration Set attribute `orclLDAPconnTimeOut`.

Password Policy Schema Elements

The `pwdPolicy` object class is an auxiliary object class containing the password policy information for a set of users in a given DIT. It contains attributes that define the password policy information for the entire directory.

Table B–23 lists and describes the attributes of the `pwdPolicy` object class. The default value for each of these attributes is 0 (zero). These attributes are single-valued, except `orclpwdIllegalValues`, which is multi-valued.

Table B–23 Attributes of the `pwdPolicy` Object Class

Attribute	Policy	Description
`orclpwdAlphaNumeric`	Number of Numeric Characters in Password	Number of numeric characters required in a password. By default, one numeric character is required. That is, the default value is 1.
`orclpwdencryptionenable`	Enable reversible user password encryption	If the value is `TRUE`, then the user password is stored in reversible encrypted form.
`orclpwdIllegalValues`	Illegal Values	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.

Table B–23 (Cont.) Attributes of the `pwdPolicy` ***Object Class***

Attribute	Policy	Description
`orclpwdipmaxfailure`	IP Lockout Maximum Failure	Specify the maximum number of falied logins from a specific IP address after which the account is locked.
`orclpwdToggle`	Old Password Can Be New Password	Specification for whether a user's old password can become the new one. By default, it can. The default value is 1.
`orlcpwdiplockout`	IP Lockout	Specify whether you want to enforce account lockout for a specific IP address. A value of `TRUE` enforces the lockout. The default is `FALSE`.
`pwdAllowUserChange`	User-defined Passwords	Indicator of whether users can change their own passwords. If allowed, then users can change their passwords by using ldapmodify. If not allowed, then the directory server verifies that the user has privileges to change the password. If the user does not have the appropriate privileges, then the directory server sends the client an error message. By default, user-defined passwords are allowed.
`pwdCheckSyntax`	Check Password Syntax	Specification for whether syntax checking is enforced. If 1, then syntax checking is enforced. The default is enabled.
`pwdCheckSyntax`	Check Password Syntax	Indicator of whether syntax checking is enforced. If 1, then syntax checking is enforced. The default value is 1. By default, password syntax checking is turned on, and user passwords must contain one numeric character.
`orclpwdpolicyenable`	Enable/disable Password Policy	Enalbed=`1` Disabled=`0`

Table B–23 (Cont.) Attributes of the `pwdPolicy` ***Object Class***

Attribute	Policy	Description
`pwdExpireWarning`	Password Expiration Warning	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 a warning 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. Example: If `pwdMaxAge` is 7200. and `pwdExpireWarning` is 3600, then your password expires after 2 hours. If you bind during the last hour, then you receive a warning that your password is about to expire.
`pwdFailureCountInterval`	Password Failure Count Interval	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.
`pwdGraceLoginLimit`	Number of Grace Logins after Password Expiration	Maximum number of grace logins allowed after a password expires. By default, no grace logins.are allowed. The default value is 3.
`pwdInHistory`	Number of Password History	How many of a user's previous passwords the directory server is to store. If a user attempts to reuse one of the passwords the directory server has stored, then the password is rejected. The directory server does not maintain a password history by default.

Table B–23 (Cont.) Attributes of the `pwdPolicy` ***Object Class***

Attribute	Policy	Description
`pwdLockout`	Password Lockout	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.
`pwdLockoutDuration`	Lockout Duration	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.
`pwdMaxAge`	Password Expiry Time	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.

Table B–23 (Cont.) Attributes of the `pwdPolicy` Object Class

Attribute	Policy	Description
`pwdMaxFailure`	Password Maximum Failure	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.
`pwdMinLength`	Minimum Number of Characters of Password	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`.
`pwdMustChange`	Password Change after Reset	Indicator of whether users must change their passwords after the first login, or after the password is reset by the administrator. Enabling this option requires users to change their passwords even if user-defined passwords are disabled. By default, users need not change their passwords after reset.

See Also: "Overview: Establishing a Password Policy for an Identity Management Realm" on page 15-4

In addition to the `pwdpolicysubentry` mentioned earlier, the object class `top` contains these operational attributes to maintain the user-password state information for each user entry.

Table B–24 Password Policy Operational Attributes of the `Top` Object Class

Attribute	Description
`orclrevpwd`	Reversible encrypted value of the user password. This attribute is generated only if the attribute `orclpwdencryptionenable` in the password policy entry is set to `TRUE`. The `orclrevpwd` attribute can be queried only by using the SSL one-way and two-way authentication mechanisms. This attribute cannot be queried over non-SSL sessions. See Also: "Storing and Managing Password Verifiers for Authenticating to Oracle Internet Directory" on page 16-2

Table B–24 (Cont.) Password Policy Operational Attributes of the `Top` *Object Class*

Attribute	Description
`orclpwdipaccountlockedtime`	The time at which a user was locked out of a specific IP address
`orclpwdlastlogintime`	The timestamp of the last login by the user
`pwdAccountLockedTime`	The time at which the user account was locked
`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
`pwdGraceUseTime`	The time stamps of each grace login by the user
`pwdHistory`	A history of user's previously used passwords
`pwdReset`	Indicator that the password has been reset and must be changed by the user on first authentication

See Also: "Overview: Establishing a Password Policy for an Identity Management Realm" on page 15-4

Password Verifier Schema Elements

Both the directory and Oracle components store the user password in the user entry, but in different attributes. Whereas the directory stores user passwords in the `userPassword` attribute, Oracle components store user password verifiers in the `authPassword`, `orclPasswordVerifier`, or `orclpassword` attribute. Table B–25 on page B-31 describes each of the attributes used by Oracle components.

Table B–25 Attributes for Storing Password Verifiers in User Entries

Attribute	Description
authPassword	Attribute for storing a password to an Oracle component when that password is the same as that used to authenticate the user to the directory, namely, userpassword. The value in this attribute is synchronized with that in the userpassword attribute. 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 attribute is modified, then the authpasswords for all applications are regenerated.
orclPasswordVerifier	Attribute for storing a password to an Oracle component when that password is different from that used to authenticate the user to the directory, namely, userpassword. The value in this attribute is not synchronized with that in the userpassword attribute. Like authPassword, this attribute is multivalued and can contain all the other verifiers that different applications use for this user's clear text password.
orclPassword	Attribute for storing only the 03LOGON verifier for enterprise users. The 03LOGON verifier is synchronized with the userpassword attribute, and it is generated by default for all user entries associated with the orcluserv2 object class. When Oracle Internet Directory is installed, a database security profile entry is created by default in the Root Oracle Context. The presence of this entry triggers the generation of 03LOGON verifiers for user entries associated with the orcluserv2 object class.

Each of these attribute types has appID as an attribute subtype. This attribute subtype uniquely identifies a particular application. For example, the appID can be the ORCLGUID of the application entry. This attribute subtype is generated during application installation.

Plug-in Schema Elements

The orclPluginConfig object class is a structural object class that must be associated with all plug-in entries. Its superclass is `top`. Table B–26 lists and describes its attributes.

Table B–26 Plug-in Attribute Names and Values

Attribute Name	Attribute Value	Mandatory?
`Cn`	Plug-in entry name	Yes
`orclPluginAttributeList`	A semicolon-separated attribute name list that controls whether the plug-in takes effect. If the target attribute is included in the list, the plug-in is invoked.	No
`orclPluginEnable`	0 = disable (default) 1 = enable	No
`orclPluginEntryProperties`	An ldap search filter type value need to be specified here. For example, if we specify `orclPluginEntryProperties:(&(objectclass=inetorgperson)(sn=Cezanne))`, then plug-in will not be invoked if the target entry has `objectclass` equal to `inetorgperson` and `sn` equal to `Cezanne`.	No
`orclPluginIsReplace`	For WHEN timing plug-in only `0` = disable (default) `1` = enable	No
`orclPluginKind`	PL/SQL	No

`orclPluginLDAPOperation`	One of the following values:
	`ldapcompare`
	`ldapmodify`
	`ldapbind`
	`ldapadd`
	`ldapdelete`
	`ldapsearch`
		Yes

`orclPluginName`	Plug-in package name	Yes

Table B–26 (Cont.) Plug-in Attribute Names and Values

Attribute Name	Attribute Value	Mandatory?
orclPluginRequestGroup	A semicolon-separated group list that controls if the plug-in takes effect. You can use this group to specify who can actually invoke the plug-in. For example, if you specify orclpluginrequestgroup:cn=security,cn=groups,dc=oracle,dc=com, when you register the plug-in, then the plug-in will not be invoked unless the ldap request comes from the person who belongs to the group cn=security,cn=groups,dc=oracle,dc=com.	No
orclPluginRequestNegGroup	A semicolon-separated group list that controls if the plug-in takes effect. You can use this group to specify who can NOT invoke the plug-in. For example, if you specify orclpluginrequestgroup: cn=security,cn=groups,dc=oracle,dc=com, when you register the plug-in, then the plug-in will not be invoked if the ldap request comes from the person who belongs to the group cn=security,cn=groups,dc=oracle,dc=com.	No
orclPluginResultCode	An integer value to specify the ldap result code. If this value is specified, then plug-in will be invoked only if the ldap operation is in that result code scenario. This is only for the POST plug-in type.	No
orclPluginShareLibLocation	File location of the dynamic linking library. If this value is not present, then Oracle Internet Directory server assumes the plug-in language is PL/SQL.	No

orclPluginSubscriberDNList	A semicolon-separated DN list that controls if the plug-in takes effect. For example:
	
	orclPluginSubscriberDNList=
	dc=COM,c=us;
	dc=us,dc=oracle,dc=com;
	dc=org,dc=us;
	o=IMC,c=US
	
	If the target DN of an LDAP operation is included in the list, then the plug-in is invoked.
		No

Table B–26 (Cont.) Plug-in Attribute Names and Values

Attribute Name	Attribute Value	Mandatory?
orclPluginTiming	One of the following values: pre when post **See Also:** "About Directory Server Plug-ins" on page 45-2 for explanations of these values	No
orclPluginType	One of the following values: operational attribute password_policy syntax matchingrule **See Also:** The chapter about the Oracle Internet Directory server plug-in framework in *Oracle Internet Directory Application Developer's Guide*	Yes
orclPluginVersion	Supported plug-in version number	No

Resource Information Schema Elements

This section lists and describes the attributes for:

- Resource access descriptors (RADs)
- Resource type information

The resource access descriptor object contains the attributes listed and described in.

Table B–27 Resource Access Descriptor (RAD) Attributes

Attribute	Description
orclResourceName	Specifies the name of the resource for which the connection information is being maintained.

Table B–27 (Cont.) Resource Access Descriptor (RAD) Attributes

Attribute	Description
orclOwnerGlobalID	Specifies the user or a group for which the preferences are being stored. The value of the attribute is same as the GUID (orclGlobalID) attribute value in the user or group entry. This attribute helps in abstracting the self-administrative access policies as a generic policy and also for querying the preferences given a user's GUID. For example, suppose that user John Doe from Acme Corporation needs to store his extended preferences. His actual user entry contains mostly white-pages information about the user and his authentication credentials. The user entry additionally has orclGUID as one of the attributes to uniquely identify him. The same orclGUID attribute value is used to populate orclOwnerGlobalID attribute while storing his resource access information. At runtime, all applications know the global identifier of John Doe, and they can easily query the directory for all his preference values.
orclApplicationGUID	Specifies the global identifier of the application entity for which the user-preferences are being stored. The value of the attribute is same as the GUID (orclGUID) attribute value for the application entity. This attribute is useful when application-specific resource access information for a user is stored under the user's container object as shown in Figure 2–10 on page 2-38.
orclResourceTypeName	Specifies the name of the resource—for example, database, XMLPDS, JDBCPDS
displayName	Specifies the display name associated with the resource
description	Specifies the description associated with orclResourceTypeName.
orclUserIDAttribute	Specifies the user identifier value to access the resource.
orclPasswordAttribute	Specifies the password value to access the resource.
orclFlexAttribute1	Specifies the additional information if required by the resource type.
orclFlexAttribute2	Specifies the additional information if required by the resource type.
orclFlexAttribute3	Specifies the additional information if required by the resource type.
OrclUserModifiable	Specifies if the data is modifiable by the user that this RAD entry is created for

Table B–28 Attributes for Resource Type Information

Attribute	Description
orclResourceTypeName	Specifies the name of the resource—for example, database, XMLPDS, JDBCPDS
displayName	Specifies the display name associated with the orclResourceTypeName
description	Specifies the description associated with orclResourceTypeName
javaClassName	Specifies the fully qualified class name used by the product to perform user authentication—DBAuth, XMLPDSAuth, JDBCPDSAuth
orclUserIDAttribute	Specifies the user identifier attribute in the encoded resource access data.
orclPasswordAttribute	Specifies the password attribute in the encoded resource access data.
orclConnectionFormat	Specifies the format used to construct the connect string associated with the resource.
OrclFlexAttribute1	Specifies the GUL label for storing extra information if required for a particular resource type.
OrclFlexAttribute2	Specifies the GUL label for storing extra information if required for a particular resource type.
OrclFlexAttribute3	Specifies the GUL label for storing extra information if required for a particular resource type.

Replication Schema Elements

Table B–29 Replication Schema Elements

Object Classes	Attributes
changeLogEntry, changeStatusEntry, orclReplAgreementEntry	orclGUID, changeNumber changeType, changes, orclParentGUID,server, changeLog, changeStatus, orclChangeRetryCount, orclAgreementId,orclReplicationProtocol, orclUpdateSchedule,targetDN, orclIncludedNamingcontexts, orclExcludedNamingcontexts, orclDirReplGroupDSAs, orclExcludedAttributes, orclreplicaDN

Note: In this release, you cannot use the `targetDN` attribute as a filter. If you do, the operation will fail.

Replication Server Configuration Parameters

Table B–30 lists and describes the attributes of the replication server configuration set entry, which has the following DN: `cn=configset0,cn=osdrepld,cn=subconfigsubentry`.

Table B–30 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
`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

See Also: "Viewing and Modifying Directory Replication Server Configuration Parameters" on page 25-36

Replica Subentry Attributes

Table B–31 Attributes of the Replica Subentry

Attribute	Description
`OrclReplicaID`	Naming attribute for the replica subentry. Its value is unique to each directory server node that is initialized at installation. The value of this attribute, assigned during installation, is unique to each directory node, and matches that of the `orclreplicaID` attribute at the root DSE. You cannot modify this value.
`orclReplicaURI`	Contains information in ldapURI format that can be used to open a connection to this replica.
`orclReplicaSecondaryURI`	Contains the set of ldapURI format addresses that can be used if the `orclReplicaURI` values cannot be used.
`orclReplicaType`	Defines the type of replica such as read-only or read/write. Possible values: ■ 0 (Read/Write) ■ 1 (Read-Only)
`orclReplicaState`	Defines the state of the replica such as bootstrap, online, and so on. Possible values: ■ 0 (Boot Strapping) ■ 1 (On-line) ■ 2 (Off-line)
`OrclReplicaVersion`	Oracle Internet Directory version of the replica.

See Also: "The Replica Subentry" on page 24-14

Replication Agreement Entry Attributes

Table B–32 Attributes of the Replication Agreement Entry

Attribute	Description
`orclagreementID`	Naming attribute for the replication agreement entry. You cannot modify this attribute.

Table B–32 (Cont.) Attributes of the Replication Agreement Entry

Attribute	Description
`OrclReplicaDN`	For LDAP-based replication only. It is required to specify the DN of the replica to identify a consumer in the replication agreement. You cannot modify this attribute.
`OrclReplicationPortocol`	Define the replication protocol for change propagation to replica. Values: ■ ODS_ASR_1.0 (Oracle9*i* Advanced Replication-based protocol) ■ ODS_LDAP_1.0 (LDAP-based replication) You cannot modify this attribute.
`OrclDirReplGroupDSAs`	For Oracle9*i* Advanced Replication-based groups, the `orclreplicaid` values of all the nodes in this replication group. This list must be identical on all nodes in the group. You can modify this attribute. This attribute is not applicable for LDAP-based agreement.
`OrclUpdateSchedule`	Replication update interval for new changes and those being retried. The value is in minutes. You can modify this attribute.
`OrclHIQSchedule`	The interval, in minutes, at which the directory replication server repeats the change application process. You can modify this attribute.
`OrclLDAPConnKeepAlive`	Attribute determining whether the connections from the directory replication server to the directory server is kept active or established every time the changelog processing is done based on various schedules. You can modify this field.
`Orcllastappliedchangenumber`	This attribute indicates the status of the consumer replica with respect to the supplier in an LDAP-based replication agreement. This attribute is not applicable to Oracle9*i* Advanced Replication-based agreements. You cannot modify this attribute.
`orclexcludednamingcontexts`	For Oracle9*i* Advanced Replication-based agreements, the value for this multivalued attribute specifies one or more subtrees to be excluded from replication. You can modify this attribute.

See Also: "The Replication Agreement Entry" on page 24-14

Replication Naming Context Objects

The container for replication naming context objects is an entry with the RDN `cn=replication namecontext`. It is created below the `orclagreementID` entry at installation. The `cn=replication namecontext` entry has the attributes listed and described in Table B–33.

Table B–33 Attributes of the Replication Naming Context Entry

Attribute	Description
`orclincludednamingcontexts`	The naming context included in a partial replica. This is a single valued attribute. For each naming context object, you can specify only one unique subtree. In partial replication, except for subtrees listed in the `orclexcluednamingcontexts` attribute, all subtrees in the specified included naming context are replicated. **Note:** Only LDAP-based replication agreements respect this attribute to define one or more partial replicas. If this attribute contains any values in an Oracle9*i* Advanced Replication-based replication agreement, then it is ignored. You can modify this attribute.
`orclexcludednamingcontexts`	In LDAP-based replication, the value for this attribute specifies the root of a subtree, located within the included naming context, to be excluded from replication. This is a multivalued attribute. From within the naming context specified in the `orclincludednamingcontexts` attribute, you can specify one or more subtrees to be excluded from the partial replication. You can modify this attribute.
`orclexcludedattributes`	Within the included naming context, an attribute to be excluded from replication. This is a multivalued attribute. **Note:** This attribute is for partial replication only.

SSL Schema Elements

Note: These attribute values are stored as part of configuration entries.

The SSL attributes are: `orclsslAuthentication, orclsslEnable, orclsslWalletURL, orclsslPort, orclsslVersion`

See Also:

- "Setting Debug Logging Levels by Using the OID Control Utility" on page 10-6 for information on debug levels
- *Oracle Advanced Security Administrator's Guide* for information on setting the location of the Oracle Wallet and the Oracle Wallet password

System Operational Attributes

The following system operational attributes are modifiable.

Table B–34 Modifiable System Operational Attributes

Attribute	Description
`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. There is no default.
`orclCryptoScheme`	Hash algorithm for encrypting the password. Options are: ■ MD4 ■ MD5 ■ No encryption ■ SHA ■ SSHA ■ UNIX Crypt The default is MD4.
`orclSizeLimit`	Maximum number of entries to be returned by a search
`orclServerMode`	Specification as to whether data can be written to the server. Valid values are read-only and read-write. The default is read-write.

Table B–34 (Cont.) Modifiable System Operational Attributes

Attribute	Description
`orclTimeLimit`	Maximum amount of time, in seconds, allowed for a search to be completed. The default is 3600.
`orclecacheenabled`	Specification as to whether entry caching, described in "Entry Caching" on page 2-3, is enabled. The value for enabled is 1; the value for disabled is `0`. The default is 1.

`orclecachemaxentrysize`	Maximum size in bytes of the entry that can be cached in the entry cache. Any entry with size greater than `orclecachemaxentrysize` is not cached. If you have an entry with many binary attributes, or `member` or `uniquemember` attributes, and need to cache, then increase `orclecachemaxentrysize` to the appropriate value.
	The default is 1 MB
	This attribute is in the entry `cn=dsaconfig,cn=configsets,cn=oracle internet directory`.
	To change this value:
	`ldapmodify -p port -D cn=orcladmin -w adminpassword << EOF`
	`dn: cn=dsaconfig,cn=configsets,cn=oracle internet directory`
	`changetype: modify`
	`replace: orclecachemaxentrysize`
	`orclecachemaxentrysize: new_integer_value`
	`EOF`

`orclecachemaxsize`	Maximum number of bytes of RAM that the entry cache can use. The default is 100M.
`orclecachemaxentries`	Maximum number of entries that can be present in the entry cache. The default is 25,000.
`orclDIPRepository`	Used by the directory replication server, and indicates whether change logs are to be generated in the consumer node for the Oracle directory integration and provisioning server to consume. The default is FALSE.
`orclEnableGroupCache`	The cache of privilege groups and ACL groups in the directory server. Using this cache improves the performance of access control evaluation for users when privilege and ACP groups are used in ACI. Use the group cache when a privilege group membership does not change frequently. If a privilege group membership does change frequently, then it is best to turn off the group cache. This is because, in such a case, computing a group cache increases overhead. The default is 1.

Table B–34 (Cont.) Modifiable System Operational Attributes

Attribute	Description
orclMatchDNEnabled	If the base DN of a search request is not found, then the directory server returns the nearest DN that matches the specified base DN. Whether the directory server tries to find the nearest match DN is controlled by this attribute. If set to 1, then match DN processing is enabled. If set to 0, then match DN processing is disabled. The default is 1.
Orclanonymousbindsflag	Specification as to whether anonymous binds are allowed or not. If set to 1, then anonymous binds are allowed. If set to 0 (zero), then they are not allowed. The default is 1.
orclStatsPeriodicity	Specification as to how often you want to gather sample statistics—that is, the number of minutes in the interval. Set this to 1 or more minutes. The default is 60.
orclStatsFlag	Indicates 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. The default is 0.
orclLDAPconnTimeOut	Specifies maximum connection time in minutes for an idle LDAP connection to be closed by the directory server. This is a DSA configuration set (DN: "cn=dsaconfig,cn=configsets,cn=oracle internet directory") attribute and its value can be set by using ldapmodify. The default is 0.
OrclEventLevel	Specifies critical events related to security and system resources that you want recorded. The default is 0—that is, no critical events are recorded Please note that for events other than super user, proxy and replication login, the value of the orclStatsFlag attribute also must be set to 1 for enabling this feature. **See Also:** "Configuring Critical Events" on page 10-22 for a list of critical events that can be monitored

Note: If you have multiple directory server instances connecting to the same database, or multiple server processes in the same directory server instance, then entry caching is automatically disabled. This is irrespective of the value of the orclecacheenabled attribute.

See Also: "Setting System Operational Attributes" on page 5-9

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 enables 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

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

Numeric String

Boolean

Object Class Description

Certificate

Octet String

Directory String

OID

DN

Presentation Address

Facsimile Telephone Number

Printable String

INTEGER

Telephone Number

JPEG

UTC Time

Name And Optional UID

Additional LDAP Syntax Recognized by Oracle Internet Directory

In addition to the commonly used LDAP syntax defined in the previous section, 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.`

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 by using the following object classes: `OrclUser`, `OrclUserV2`, in addition to `inetOrgPerson`. Table B–35 describes the attribute names.

Table B–35 User Attributes

Attribute Name	Mandatory or Optional	Description
`OrclGUID`	Optional	Specifies a Unique Global ID to identify the user.
`Cn`	Mandatory	Specifies user's first name, common nickname, or both.
`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 birth date, 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.

Table B–35 (Cont.) User Attributes

Attribute Name	Mandatory or Optional	Description
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.
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.
FacsimileTelephone Number	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) format. If the attribute is missing, then 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.
OrclPasswordHint Answer	Optional	Specifies the answer to the password hint question.

Table B–35 (Cont.) User Attributes

Attribute Name	Mandatory or Optional	Description
`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.
`OrclTimeZone`	Optional	Specifies the time zone applicable for a user location.
`OrclDefaultProfile Group`	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.
`OrclWorkflow Notification Preference`	Optional	Specifies the preferred delivery mechanism for sending workflow notification to a user.

C

Elements in Oracle Internet Directory Graphical User Interfaces

This appendix lists and describes the various fields and control devices in Oracle Directory Manager and the Oracle Internet Directory Self-Service Console. It contains these topics:

Fields in Oracle Directory Manager

This section contains these topics:

Access Control Management Fields in Oracle Directory Manager

Table C–1 Fields in the Access Control Management Pane

Field	Description
Path to the Subtree Control Point	Contains the path defined by the ACP.
Subtree Control Point	Contains the ACP

Table C–2 lists and describes the authentication choices—that is, the methods by which users can be authenticated to the directory.

Table C–2 Fields in Authentication Choice List

Authentication Choice	Description
MD5Digest.	Binding by using MD5Digest blocks Simple, Proxy and Anonymous access.
PKCS12	Binding by using PKCS12 blocks MD5Digest, Simple, Proxy and Anonymous access

Table C–2 (Cont.) Fields in Authentication Choice List

Authentication Choice	Description
Proxy	▪ Binding as a proxy user. Specifying this authentication option blocks anonymous access.
Simple	▪ Password-based authentication. Specifying this option blocks both Proxy and Anonymous access.

Table C–3 lists and describes the encryption choices—that is, the method by which data is encrypted.

Table C–3 Fields in Encryption Choice List

Authentication Choice	Description
SASL	Simple Authentication and Security Layer
SSL No 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 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.

See Also: Bind Mode on page 14-10

Table C–4 Entities to Whom You Are Granting Access in the By Whom Tab Page

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 Group Is Identified By Attribute	Any group whose DN is an attribute in the entry.

Table C–4 (Cont.) Entities to Whom You Are Granting Access in the By Whom Tab

Entity	Description
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

Table C–5 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

Attribute Uniqueness Fields in Oracle Directory Manager

Table C–6 Fields in the New Constraint Dialog Box

Field	Description
Attribute Uniqueness Constraint Name	Name of the attribute uniqueness constraint you are creating
Unique Attribute Name	The attribute you want the directory server to check
Unique Attribute Object Class	The object class where the attribute uniqueness constraint is enforced—for example, person. By default, it is enforced on all object classes.

Table C–6 Fields in the New Constraint Dialog Box

Field	Description
Unique Attribute Scope	The filter you want the directory server to use when searching for an attribute constraint. For example: ■ base—Searches the root entry only ■ onelevel—Searches one level only ■ sub—Searches the entire directory
Unique Attribute Subtree	The subtree where the attribute uniqueness constraint is enforced. By default, it is enforced from the root directory.

Garbage Collection Management Fields in Oracle Directory Manager

Table C–7 Fields in the Garbage Collector Window

Field	Description
Garbage Collector Name	You cannot modify this field.
Purge Base	The base DN of the naming context to which the garbage collection task is to be applied. You cannot modify this field.
Purge Debug	Indicator of whether to enable or disable debug logging for this garbage collector
Purge Enable Status	Enable or disable this garbage collector. The default is Enable.
Purge File Location	Absolute path name of the directory in which the log file is located
Purge File Name	Name of the log file
Purge Interval	The interval, in hours, after which the Garbage Collection job is executed again. For example, if you set this value to 12, then garbage collection occurs every 12 hours. This attribute is optional. The default value is `24`.
Purge Now	Entering any value in this field means that, when you choose Apply, the garbage collection begins immediately. At that point, the value in this field automatically reverts to null.
Purge Start	Time, in seconds, when the Garbage collector runs for the first time. The format is *`YYYYMMDDHH24MISS`*. This attribute is optional. The default value is 0, which means that the garbage collector is enabled immediately.

Table C–7 (Cont.) Fields in the Garbage Collector Window

Field	Description
Purge Target Age	Age, in hours, of the target objects. Objects older than the age specified in this attribute are purged at midnight. This attribute is optional. The default value is 12.
Purge Transaction Size	Number of objects to be purge in one committed transaction. This attribute is optional. The default value is `1000`.

Password Policy Fields in Oracle Directory Manager

Table C–8 Fields in the Password Policies General Tab Page

Field	Description
Enable OID Password Policy	To disable the default Oracle Internet Directory password policy, select Disable. The default is Enable.
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.
Number of Grace Logins after Password Expiration	Maximum number of grace logins allowed after a password expires. By default, no grace logins.are allowed. The default value is 3.

Table C–8 (Cont.) Fields in the Password Policies General Tab Page

Field	Description
Password Expiration Warning	Enter the number of seconds in which users must modify their passwords before those passwords expire. The directory server sends a password expiration warning if these two conditions are met: ■ The attribute for the expiry time for a user's password is set ■ This attribute is also enabled From that point, the user has a specified number of seconds in which to modify the password. If the user does not modify the password within the specified number of seconds, then the password expires and the user is locked out until the password is changed by the administrator. For example, suppose that: ■ The Password Expiry Time is set to `7200`—that is, your password expires after 2 hours ■ The Password Expiration Warning is set to `3600`—that is, 1 hour In this example, if you bind during the last hour, then you receive a warning that your password is about to expire. If you do not modify your password during that time, then your password expires and you are locked out of your account until the administrator changes your password. For this feature to work, the client application must support it. The default is `0`, which means no warnings are sent.
Password Expiry Time	Enter the number of seconds that a given password is valid. For example, if you set the value of this attribute to `7200`, then the password expires in two hours from the time that you set it. If this attribute is not present, or if the value is 0, then the password does not expire. By default, passwords expire in 60 days.
Password Policy Entry	This field displays the RDN of the password policy entry. You cannot edit this field.
Path to Password Policy Entry	This field displays the full DN of the password policy entry. You cannot edit this field.

Table C–9 Fields in the Password Policies Account Lockout Tab Page

Field	Description
Global Lockout Duration	Enter the number of seconds a user is locked out of the global directory if both of the following are true: ■ Global 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. The default value is 24 hours.
Password Failure Count Interval	Enter the number of seconds after which the password failure times are purged from the user entry.
Password Maximum Failure	Enter the number of consecutive failed bind attempts after which a user account is locked.

Table C–10 Fields in the Password Policies IP Lockout Tab Page

Field	Description
IP Lockout Duration	Specify the number of seconds you want to enforce account lockout for a specific IP address.
IP Lockout Maximum Failure	Specify the maximum number of failed logins from a specific IP address after which the account is locked.

Table C–11 Fields in the Password Policies Password Syntax Tab Page

Field	Description
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.
Number of Password History	Specify how many of a user's previous passwords the directory server is to store. If a user attempts to reuse one of the passwords the directory server has stored, then the password is rejected. The directory server does not maintain a password history by default.

Table C–11 (Cont.) Fields in the Password Policies Password Syntax Tab Page

Field	Description
Password Illegal Values	Enter the common words and attribute types whose values cannot be used as a valid password. By default, all words are acceptable password values.

Password Verifier Fields in Oracle Directory Manager

Table C–12 Fields in the 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.
Owner	The DN of the administrator of the verifier entry. You can modify this field.
Application ID	The unique identifier of the Oracle application. It is generated during application installation. You cannot modify this field.

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`

Plug-in Management Fields in Oracle Directory Manager

See Also: "Registering and Managing Plug-ins" on page 45-4

Table C–13 New Plug-in Dialog Box

Field	Description
Mandatory Properties Tab Page	
Plug-in Entry Name	For example, cn=my_plugin. This field is mandatory.
Plug-in Kind	PL/SQL. This field is mandatory.
Plug-in LDAP Operation	One of the following values: ■ ldapcompare ■ ldapmodify ■ ldapbind ■ ldapadd ■ ldapdelete ■ ldapsearch This field is mandatory.
Plug-in Package Name	This field is mandatory.

Table C–13 (Cont.) New Plug-in Dialog Box

Field	Description
Plug-in Type	One of the following values: ■ `operational`--Operation plug-ins augment existing LDAP operations. The work they perform depends on whether they execute before, after, or in addition to normal directory server operations. ■ `attribute`--Attribute-based plug-ins involve additions or changes to attribute values, and perform tasks that are in addition to various LDAP operations related to those attributes. For example, suppose that, whenever a credit card number is added to the directory, you want to encrypt it. To do this, you can create a plug-in for the credit card number attribute, and specify that, before any credit card number is added, the plug-in is called to encrypt it. Similarly, you can specify that, after a credit card number is retrieved in a search, the plug-in is called to decrypt it. ■ `replacement`--Every LDAP operation consists a sequence of modules. For example, the ldapmodify operation has a module for attribute value checking, one for schema checking, one for ACL evaluation, and one for directory modification. The replacement plug-in enables you to replace a module, customizing the operation as needed. If this type of plug-in module fails, then the associated LDAP operation fails as well. This field is mandatory. **See Also:** Chapter 45, "Oracle Internet Directory Plug-in Framework"
Optional Properties Tab Page	
Plug-in Enable	Acceptable values are: ■ 0 = disable (default) ■ 1 = enable This attribute is optional.
Plug-in Entry Properties	An LDAP search filter type. For example, if you specify `orclPluginEntryProperties:(&(objectclass=inetorgperson)(sn=Cezanne))`, then the plug-in will not be invoked if the target entry has `objectclass` equal to `inetorgperson` and `sn` equal to `Cezanne`.

Table C–13 (Cont.) New Plug-in Dialog Box

Field	Description
Plug-in Replacement	For WHEN timing plug-in only. Possible values are: ■ Disable (default) ■ Enable This property can be enabled only if the Plug-in LDAP Operation property is ldapbind, ldapcompare, or ldapmodify. This attribute is optional.
Plug-in Request Group	A group list that controls if the plug-in takes effect. You can use this group to specify who can actually invoke the plug-in. For example, if you specify `cn=security,cn=groups,dc=oracle,dc=com, then,` when you register the plug-in, the plug-in will not be invoked unless the LDAP request comes from a member of the group `cn=security,cn=groups,dc=oracle,dc=com`.
Plug-in Result Code	An integer value to specify the LDAP result code. If this value is specified, then plug-in will be invoked only if the LDAP operation is in that result code scenario. This is only for the POST plug-in type.
Plug-in Subscriber DN List	A semicolon separated DN list that controls if the plug-in takes effect. For example: `orclPluginSubscriberDNList=dc=COM,c=us; dc=us,dc=oracle,dc=com;dc=org,dc=us;o=IMC,c=US` The target DN of an LDAP operation is included in the list, then the plug-in is invoked.
Plug-in Timing	One of the following values: ■ pre--for plug-ins that the directory server calls *before* performing an LDAP operation ■ `when`--for plug-ins that the directory server calls in addition to standard processing of an LDAP operation ■ `post`--for plug-ins that the directory server calls after performing an LDAP operation This attribute is optional.
Plug-in Version	Supported plug-in version number. This attribute is optional.

Replication Fields in Oracle Directory Manager

Table C–14 Fields in the Replication Server Configuration Set: General Tab Page

Field	Description
Change Retry Count	Enter 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`. You can modify this field.
Number of Threads Per Supplier	Enter the number of worker threads the directory replication server provides for each supplier for change log processing. The default is `5`. You can modify this field.

See Also: "Modifying Configuration Parameters of the Directory Replication Server by Using Oracle Directory Manager" on page 25-36

Table C–15 Fields in the ASR Agreement Tab Page

Field	Description
Excluded Naming Contexts	The root of a subtree to be excluded from replication. This is a multivalued attribute. You can modify this field.
HIQ Schedule	The interval, in minutes, at which the directory replication server repeats the change application process. You can modify this field.
Keep LDAP Connection Alive	This attribute determines whether connections from the directory replication server to the directory server are kept active or established every time the changelog processing is done based on various schedules. You can modify this field.
Replica Agreement ID	Naming attribute for the replication agreement entry.
Replica Agreement Protocol	This attribute defines the replication protocol for change propagation to the replica. Values: ▪ `ODS_ASR_1.0` (Oracle9*i* Advanced Replication-based replication) ▪ `ODS_LDAP_1.0` (LDAP-based replication)

Table C–15 (Cont.) Fields in the ASR Agreement Tab Page

Field	Description
Replication Group Nodes	For Oracle9*i* Advanced Replication-based groups, enter the `orclreplicaid` values of all the nodes in this replication group. This list must be identical on all nodes in the group. This attribute is not applicable to LDAP-based replication agreements.
Update Schedule	Replication update interval for new changes and those being retried. The value is in minutes. You can modify this field.

Table C–16 Fields in the Replica Node: General Tab Page

Attribute	Description
Replica ID	Naming attribute for the replica subentry. Its value is unique to each directory server node that is initialized at installation. The value of this attribute, assigned during installation, is unique to each directory node, and matches that of the `orclreplicaID` attribute at the root DSE. You cannot modify this value.
Replica Secondary URI	Contains the set of ldapURI format addresses that can be used if the `orclReplicaURI` values cannot be used.
Replica State	Defines the state of the replica such as bootstrap, online, and so on. Possible values: ■ 0 (Boot Strapping) ■ 1 (On-line) ■ 2 (Off-line)
Replica Type	Defines the type of replica such as read-only or read/write. Possible values: ■ 0 (Read/Write) ■ 1 (Read-Only)
Replica URI	Contains information in `ldapURI` format that can be used to open a connection to this replica
See Also	DN of the infrastructure database used by Oracle Internet Directory. This field is not modifiable.

Table C–17 Columns in the Replica Agreements Tab Page

Column	Description
Consumer Replica DN	This attribute specifies the DN of the replica to identify a consumer in the replication agreement. You can modify this field.
HIQ Schedule	The interval, in minutes, at which the directory replication server repeats the change application process. You can modify this field.
Keep LDAP Connection Alive	This attribute determines whether connections from the directory replication server to the directory server are kept active or established every time the changelog processing is done based on various schedules. You can modify this field.
Last Applied Change Number	This attribute indicates the status of the consumer replica with respect to the supplier in an LDAP-based replication agreement. This attribute is not applicable for Oracle9*i* Advanced Replication-based agreements.
Replica Agreement ID	Naming attribute for the replication agreement entry.
Replication Protocol	This attribute defines the replication protocol for change propagation to the replica. Values: ■ ODS_ASR_1.0 (Oracle9*i* Advanced Replication-based replication) ■ ODS_LDAP_1.0 (LDAP-based replication)
Update Schedule	Replication update interval for new changes and those being retried. The value is in minutes. You can modify this field.

Table C–18 Fields in the Replica Naming Context Tab Page

Field	Description
Excluded Attributes	For partial replication only. Within the included naming context, an attribute to be excluded from replication. This is a multivalued attribute.

Table C–18 (Cont.) Fields in the Replica Naming Context Tab Page

Field	Description
Excluded Naming Contexts	The root of a subtree to be excluded from replication. This is a multivalued attribute. You can modify this field. For LDAP-based replication, from within the naming context specified in the `orclincludednamingcontexts` attribute, you can specify one or more subtrees in the LDAP naming context object so that they are excluded from partial replication. For replication agreements based on Oracle9*i* Advanced Replication, you can specify one or more subtrees to be excluded from replication.
Included Naming Contexts	The naming context included in a partial replica. This is a single valued attribute. For each naming context object, you can specify only one unique subtree. In partial replication, except for subtrees listed in the `orclexcluednamingcontexts` attribute, all subtrees in the specified included naming context are replicated. **Note:** Only LDAP-based replication agreements respect this attribute to define one or more partial replicas. If this attribute contains any values in an Oracle9*i* Advanced Replication-based replication agreement, then it is ignored. You can modify this attribute.

See Also: "Determining What Is to Be Replicated in LDAP-Based Partial Replication" on page 25-31

Table C–19 Fields in the Change Log Window

Field	Description
Change Log Number	The unique identifier of this change
Change Log Operation	The type of operation that this change effected--for example, add, modify, delete, compare
Change Log Target DN	The DN of the entry upon which this change was effected
Change Log Target DN Changes	The changes made to the entry
Change Retry Count	The number of attempts to apply this change to another node in a replicated environment

Table C–19 (Cont.) Fields in the Change Log Window

Field	Description
Modifier's Name	The name of the user who effected the change
Operation Time	The time at which the change took place
Orcl GUID	The global unique identifier of the entry on which the change is made
Orcl Parent GUID	The global unique identifier of the parent of the entry on which the change is made
Server Name	The name of the server from which the change was issued

Schema Management Fields in Oracle Directory Manager

See Also: Chapter 6, "Directory Schema Administration"

This section contains these topics:

Object Classes Fields in Oracle Directory Manager

Table C–20 Object Class Properties Listed in Searches in Oracle Directory Manager

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. The object identifier 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.

Table C–20 (Cont.) Object Class Properties Listed in Searches in Oracle Directory

Option	Description
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. This field is optional, for your information only.
Type	Type of object class for which you are searching, whether abstract, structural, or auxiliary
Super Class	Class from which the object class for which you are searching is derived. Clicking Add displays the Super Class Selector dialog box from which you can select the superclass(es) you want to add.
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

Table C–21 Search Filters for Object Classes

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.
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.

Table C–21 (Cont.) Search Filters for Object Classes

Filter	Description
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.

Table C–22 Buttons Used in Searches for Object Classes in Oracle Directory Manager

Button	Description
New	Creates a new search criteria bar in the **Criteria** field. This button is enabled only when the **Criteria** bar has been deleted.
And	Creates another search criteria bar in the **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 **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

Table C–23 Fields in the New Object Class Dialog Box

Option	Description
Name	Name of the object class.
Object ID	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	Type of object class: Abstract, Structural, Auxiliary, None.

Table C–23 (Cont.) Fields in the New Object Class Dialog Box

Option	Description
Super Class	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	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	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.

Attributes Fields in Oracle Directory Manager

Table C–24 Columns in the Attributes Tab Page in Oracle Directory Manager

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 each attribute
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` ▪ `dSAOperation`.
Ordering	Standards specifying how precedence is established for values
Equality	Standards specifying how equality is determined in compare and search operations
Substring	Regular expression matching string
Single Value	Attribute types containing a maximum of one value

Table C–24 (Cont.) Columns in the Attributes Tab Page in Oracle Directory Manager

Column	Description
Super	Super attribute for each attribute

Table C–25 Search Filters for Attributes

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.

Table C–26 Buttons in Searches for Attributes in Oracle Directory Manager

Button	Description
New	Creates a new search criteria bar in the **Criteria** field. This button is enabled only when the **Criteria** field is empty.
And	Creates another search criteria bar in the **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 **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

Table C–27 Fields in the General Tab Page of the New Attribute Type Dialog

Field	Description
Name	Name for this attribute
Object ID	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 at `http://www.ietf.org`.
Description	Optional field for your information only
Syntax	Standardized rules for data entry applicable to this attribute type
Size	Maximum size allowed for this object
Single Value	Indicator that this attribute type contains a maximum of one value.

Table C–28 Fields in the Advanced Tab Page of the New Attribute Type Dialog

Field	Description
Indexed	Select this box to add the 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.

Table C–28 (Cont.) Fields in the Advanced Tab Page of the New Attribute Type Dialog

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 the matching rule.
Super	Add the super attribute for this attribute. To do this: 1. Choose the Add button next to this field. The Super Attribute Selector appears. 2. Select the super attribute and choose Select. 3. Repeat as needed. To delete a super attribute from the Super field, select it, then choose Delete.

Matching Rules Fields in Oracle Directory Manager

Table C–29 Fields in the Matching Rules Tab Page

Column Head	Description
Name	Name of the attribute matching rule
Object ID	Unique identifier of this matching rule
Description	Words describing the matching rule (optional)
Syntax	Syntax used with this matching rule

Content Rules Management Fields in Oracle Directory Manager

Table C–30 Fields in the New Content Rule Dialog Box

Field	Description
Structural Object Class	The name of the structural object class to which you want to assign this content rule
Object ID	The unique identifier of the content rule you are creating
Label	A descriptive friendly name of this content rule
Auxiliary Classes	The auxiliary object classes whose attributes you want to associate with the specified structural object class. To specify an auxiliary class: 1. Choose Add. The Auxiliary Class Selector dialog box appears. 1. Select the auxiliary class you want to add. 2. Choose Select. This returns you to the New Content Rule dialog box. The auxiliary class you just specified appears in the Auxiliary Classes field.
Mandatory Attributes	The mandatory attributes you want to associate with the specified structural object class. To specify a mandatory attribute: 1. Choose Add. The Mandatory Attribute Selector dialog box appears. 1. Select the mandatory attribute you want to add. If you want this attribute indexed, then select the corresponding check box in the Indexed column. 2. Choose Select. This returns you to the New Content Rule dialog box. The mandatory attribute you just specified appears in the Mandatory Attributes field.
Optional Attributes	The optional attributes you want to associate with the specified structural object class. To specify an optional attribute: 1. Choose Add. The Optional Attribute Selector dialog box appears. 2. Select the optional attribute you want to add. If you want this attribute indexed, then select the corresponding check box in the Indexed column. 3. Choose Select. This returns you to the New Content Rule dialog box. The optional attribute you just specified appears in the Optional Attributes field.

Table C–31 Fields in the Content Rule Dialog Box

Field	Description
Structural Object Class	The name of the structural object class to which you want to assign this content rule
Object ID	The unique identifier of the content rule you are creating
Label	A descriptive friendly name of this content rule
Auxiliary Classes	The auxiliary object classes whose attributes you want to associate with the specified structural object class. To specify an auxiliary class: 1. Choose Add. The Auxiliary Class Selector dialog box appears. 1. Select the auxiliary class you want to add. 2. Choose Select. This returns you to the New Content Rule dialog box. The auxiliary class you just specified appears in the Auxiliary Classes field.
Mandatory Attributes	The mandatory attributes you want to associate with the specified structural object class. To specify a mandatory attribute: 1. Choose Add. The Mandatory Attribute Selector dialog box appears. 1. Select the mandatory attribute you want to add. If you want this attribute indexed, then select the corresponding check box in the Indexed column. 2. Choose Select. This returns you to the New Content Rule dialog box. The mandatory attribute you just specified appears in the Mandatory Attributes field.
Optional Attributes	The optional attributes you want to associate with the specified structural object class. To specify an optional attribute: 1. Choose Add. The Optional Attribute Selector dialog box appears. 2. Select the optional attribute you want to add. If you want this attribute indexed, then select the corresponding check box in the Indexed column. 3. Choose Select. This returns you to the New Content Rule dialog box. The optional attribute you just specified appears in the Optional Attributes field.

Server Management Fields in Oracle Directory Manager

This section contains these topics:

Configuration Sets Fields in Oracle Directory Manager

See Also: Managing Server Configuration Set Entries by Using Oracle Directory Manager on page 5-4

Table C–32 Fields in the Configuration Sets Dialog Box—General Tab Page

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.
Non-SSL Port	The default non-SSL port is 389. You can change the non-SSL port.
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.
SASL Authentication Mode	The default value is 1. No other values are supported in this release of Oracle Internet Directory.
SASL Mechanism	The default value is DIGEST-MD5. No other values are supported in this release of Oracle Internet Directory.
SASL Cipher Choice	The default values for this multivalued attribute are: ■ RC4-56 ■ DES ■ 3DES

Table C–33 Fields in the Configuration Sets—SSL Settings Tab 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 Enable	Choose one of the following: ■ Both SSL and Non-SSL—Both non-secure operation and SSL authentication ■ Non-SSL Only—Only non-secure operation; default port is 389, changeable in the SSL Port field ■ SSL Only—Only SSL authentication; default port is 636, changeable in the SSL Port field

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 UNIX, 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 Port	The default SSL port is 636. You can change the SSL port.

System Operational Attributes Fields in Oracle Directory Manager

See Also: "Setting System Operational Attributes by Using Oracle Directory Manager" on page 5-9

Table C–34 System Operation Attributes Displayed in Oracle Directory Manager

Field	Description	Default Value	Modifiable?
Allow Anonymous Binds	Indicator of whether anonymous binds are allowed or not. If set to `1`, then anonymous binds are allowed. If set to 0 (zero), then they are not allowed.	`1`	Yes
Alternate Server	When connectivity to the local server is lost, clients have the option of accessing one of the servers listed in this attribute. Specify other Oracle directory servers in the system that have the same set of naming contexts as that of the local server. The format is: ldap://*host_name*:*port_number* **See Also:** "Setting the Alternate Server List by Using Oracle Directory Manager" on page 26-4	`None`	Yes
Configuration Set Location	DN of the entry holding the top of the naming context in this server	`cn=subconfigsubentry`	No
Critical Event Level	Specify critical events related to security and system resources that you want recorded. Please note that for events other than super user, proxy and replication login, the value of the `orclStatsFlag` attribute also must be set to `1` for enabling this feature. **See Also:** "Configuring Critical Events" on page 10-22 for a list of critical events that can be monitored	0	Yes
DIP Repository	Used by the directory replication server, and indicates whether change logs are to be generated in the consumer node for the Oracle directory integration and provisioning server to consume.	`FALSE`	Yes
Directory Version	The version or release of Oracle Internet Directory that you are using	9.0.4.0.0	No

Table C–34 (Cont.) System Operation Attributes Displayed in Oracle Directory Manager

Field	Description	Default Value	Modifiable?
Enable Entry Cache	Specify whether entry caching, described in "Entry Caching" on page 2-3, is enabled. The value for enabled is 1; the value for disabled is `0`.	1	Yes
Enable Group Cache	The cache of privilege groups and ACL groups in the directory server. Using this cache improves the performance of access control evaluation for users when privilege and ACP groups are used in ACI. Use the group cache when a privilege group membership does not change frequently. If a privilege group membership does change frequently, then it is best to turn off the group cache. This is because, in such a case, computing a group cache increases overhead.	`1`	Yes
Enable Match DN Processing	If the base DN of a search request is not found, then the directory server returns the nearest DN that matches the specified base DN. Whether the directory server tries to find the nearest match DN is controlled by this attribute. If set to 1, then match DN processing is enabled. If set to `0`, then match DN processing is disabled.	`1`	Yes
Enable Statistics Gathering	Indicator of 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`.	`0`	Yes
Entry Cache Size in Bytes	The maximum number of bytes of RAM that the entry cache can use.	100M	Yes
Indexed Attribute Locations	Specify the DN for the file containing all indexed attributes	`cn=catalogs`	No
Maximum Entries in Entry Cache	Specify the maximum number of entries that can be present in the entry cache.	25,000	Yes

Table C–34 (Cont.) System Operation Attributes Displayed in Oracle Directory Manager

Field	Description	Default Value	Modifiable?
Maximum TCP Connection Idle Time	Specify how long the server should keep an idle connection open before closing it.	120	
Naming Contexts	Specify the topmost DNs of naming contexts in this server that you want to publish. You must have super user privileges to publish a DN as a naming context.	None	Yes
Password Encryption	Hash algorithm for encrypting the password. Options are: ▪ MD4 Secure Hash Algorithm ▪ MD5 Secure Hash Algorithm ▪ 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
Replica ID	Unique identifier of a node in a replication agreement		
Replication Agreements	DN of the entry holding the replication agreement	`cn=orclareplagreements`	No
Replication Log Location	DN of the entry holding the change log in this server	`cn=changelog`	No
Replication Status Location	DN of the entry holding the change status in this server	`cn=changestatus`	No
Schema Definition Location	DN of the schema	`cn=subschemasubentry`	No
Server Mode	Indicator of 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, Read/Modify and Read-Only

Table C–34 (Cont.) System Operation Attributes Displayed in Oracle Directory Manager

Field	Description	Default Value	Modifiable?
Server Operation Time Limit	Maximum amount of time, in seconds, allowed for a search to be completed	3600	Yes
Simple Modify Changelog Attribute	In a multimaster replication group, resolving conflicts for changes in some attribute values can require considerable resources. You can avoid this performance degradation by specifying those attributes in this field. When you specify attributes in this field, any changes to the values of those attributes are reflected in the change log. However, in a multimaster replication group, conflict resolution for those attributes is turned off.	uniquemember member	Yes
Statistics Collection Interval	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.	`60`	Yes
Statistics Level	Specify 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`.	`0`	Yes
Supported Control List	Enter 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

Table C–34 (Cont.) System Operation Attributes Displayed in Oracle Directory Manager

Field	Description	Default Value	Modifiable?
Supported Extension	The unique identifiers of proprietary extensions to LDAP operations that are supported in this release of Oracle Internet Directory. In Release 9.0.4, there is one extended operation. It enables a plug-in using a PL/SQL package in the database to bind to the directory server.	2.16.840.1.113894.1.9.1	No
Supported LDAP Version	LDAP version that Oracle Internet Directory supports	`LDAP Version 2` `LDAP Version 3`	No
Supported SASL Mechanisms	Some clients can use the Simple Authentication and Security Layer (SASL). This field indicates the authentication mechanisms supported by the directory server. **See Also:** ▪ "How a SASL-Enabled Client Authenticates to a Directory Server by Using Digest-MD5" on page 12-9 ▪ "How a SASL-Enabled Client Authenticates to a Directory Server by Using External Authentication" on page 12-9	`DIGEST-MD5`	No
Upgrade in Progress	Reserved for upgrade	`FALSE`	No

Super, Guest, and Proxy User Fields in Oracle Directory Manager

See Also: "Managing Super Users, Guest Users, and Proxy Users by Using Oracle Directory Manager" on page 5-12

Table C–35 Fields in the System Passwords Tab Page

Field	Description
Super User Name	Type the super user name, or choose Browse to search for it. The default is `orcladmin`.

Table C–35 (Cont.) Fields in the System Passwords Tab Page

Field	Description
Super User Password	Type the super user password. The default is the same as the password you specified for the Oracle Application Server administrator (ias_admin) during installation. You should change this password immediately.
Guest Login Name	Type the guest login name, or choose Browse to search for it. Guests have privileges determined by the **access control list (ACL)** 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, or choose Browse to search for it. 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.

Query Optimization Fields in Oracle Directory Manager

See Also: "Optimizing Searches for Skewed Attributes by Using Oracle Directory Manager" on page 21-13

Table C–36 Fields in the Query Optimization Tab Page

Field	Description
Attributes with Low Cardinality	Enter the attributes you want to designate as skewed. See Also: "Optimizing Searches" on page 21-12 for a discussion of skewed attributes
Common Name	The common name of the entry containing information about skewed attributes, namely, `dsaconfig`. You cannot modify this field.
Distinguished Name	The DN of the entry containing information about skewed attributes. You cannot modify this field.
LDAP Connection Timeout	Enter the maximum number of seconds that the directory client can remain idle before the connection is terminated. The default is `0`, meaning that there is no timeout.
Maximum Entry Size in Cache	Specify the upper size limit of entries stored in the cache. The default is `5000`—that is, 5 kilobytes.

Table C–36 (Cont.) Fields in the Query Optimization Tab Page

Field	Description
Object Class	The object classes associated with the `dsaconfig` entry.
Time limit mode	When you set the server operation time limit as described in "Setting System Operational Attributes" on page 5-9, you specified the maximum number of seconds allowed for a search to be completed. In this field, to adjust server performance, set the search time limit to be either accurate or approximate. If you specify it as accurate, then searches end precisely at the specified number of seconds. If you specify it as approximate, then searches end within a few seconds of the specified number of seconds. In smaller workloads, the latter provides better performance.

Entry Search Fields and Buttons in Oracle Directory Manager

Table C–37 Search Filters for Entries

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, and so on
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, and so on.
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, and so on.
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.

Table C–37 (Cont.) Search Filters for Entries

Filter	Description
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. 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.

Table C–38 Buttons in Searches for Entries

Button	Description
New	Creates a new search criteria bar in the **Criteria** field. This button is enabled only when the **Criteria** field is empty.
And	Creates another search criteria bar in the **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 **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 6-16 ■ "Indexing an Attribute by Using Command-Line Tools" on page 6-19

SSL Management Fields in Oracle Directory Manager

See Also:

- Table C–33 on page C-28
- "Configuring SSL Parameters by Using Oracle Directory Manager" on page 13-3

Table C–39 Fields in the SSL Settings Tab 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. 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.
SSL Enable	Choose one of the following: ■ **Both SSL and non-SSL**— for both non-secure operation and SSL authentication ■ **Non-SSL only**—for non-secure operation only ■ **SSL only**—for SSL authentication only

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 UNIX, 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 Port	The default SSL port is 636. You can change the SSL port.

Synchronization Fields in Oracle Directory Manager

Fields for Registering a Directory Integration Profile

Table C–40 Fields on the General Tab Page for Synchronization 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`.
Profile Status	Specify whether the profile is enabled or disabled. This field is mandatory. The default is `ENABLE`.
Profile Password	Specify the password that directory integration and provisioning server is to use when binding to Oracle Internet Directory on behalf of the profile. This field is mandatory and the default is `welcome`.
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`.

Table C–40 (Cont.) Fields on the General Tab Page for Synchronization in Oracle Directory Manager

Field	Description
Maximum Number of Retries	Specify the maximum number of times the directory integration and provisioning 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 second retry happens 2 minutes after the second failure, and subsequently the retry takes place n minutes after the n-th failure.
Profile Version	Version of the Oracle Directory Integration and Provisioning platform with which this profile was created.

Table C–41 Fields on the Execution Tab for Synchronization in Oracle Directory Manager

Field	Description

Agent Execution Command	Specify the agent executable name and the arguments used by the directory integration and provisioning 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 path name), and
	`user=%orclodipcondirAccessAccount pass=%orclodipcondiraccesspassword`
	are the command-line 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 the Oracle Human Resources agent.

Table C–41 (Cont.) Fields on the Execution Tab for Synchronization in Oracle Directory Manager

Field	Description
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, then 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 and provisioning server passes to an agent. You cannot modify this field through Oracle Directory Manager. 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.
Interface Type	The format used by the import or export file. Options are `DB`, `LDAP`, `LDIF`, and `TAGGED`. This field is optional. The default is `TAGGED`.

Table C–42 Fields on the Mapping Tab Page for Synchronization 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"
Connected Directory Matching Filter	Specify the attribute that uniquely identifies an entry in the connected directory.

Table C–42 (Cont.) Fields on the Mapping Tab Page for Synchronization in Oracle Directory Manager

Field	Description
OID Matching Filter	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.

Table C–43 Fields on the Status Tab Page for Synchronization in Oracle Directory Manager

Field	Description
OID Last Applied Change Number (Import operations only)	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.
Synchronization Errors	The last error message. You cannot modify this field. There is no default.
Last Applied Change Number (Export operations only)	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.
Bootstrap Status	

Fields in Oracle Internet Directory Self-Service Console

This section contains these topics:

User Management Fields in the Oracle Internet Directory Self-Service Console

Table C–44 Fields in the Add New Attributes Window

Field	Description
Directory Attribute Name	The attribute name
UI Label	Specify the friendly name of the attribute to be displayed in the user interface. For example, you can display the `sn` attribute as *Last Name* in the interface.
Required Field	Specify whether you want the attribute to be required in user creation and modification. 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.
Self-Editable	Specify whether the end user can modify the value for this attribute in his or her own entry by using the Edit My Profile window.
Password Reset Validation	Select to specify that this attribute can be used to validate the user if the user forgets his or her password.
Searchable	By default, when a user enters a search request, the Oracle Internet Directory Self-Service Console searches based on the `cn`, `firstname`, `lastname`, and `e-mail` attributes. You can customize the attributes that can be searchable. For example, if you want to enable searching based on the attribute you are adding, then select this check box. The only restriction is that, to be searchable, the attribute must be cataloged.

Table C–44 (Cont.) Fields in the Add New Attributes Window

Field	Description
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. When you select this type of interface, the LOV Values text area appears. In that text area, enter the values for the list, pressing the ENTER key after each one. ■ 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

See Also: "Configuring User Entries by Using the Oracle Internet Directory Self-Service Console" on page 31-14

Table C–45 Fields in the Editing Attribute Window

Field	Description
UI Label	Specify the friendly name of the attribute to be displayed in the user interface. For example, you can display the `sn` attribute as *Last Name* in the interface.
Required Field	Specify whether you want the attribute to be required in user creation and modification. 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.
Self-Editable	Specify whether the end user can modify the value for this attribute in his or her own entry by using the Edit My Profile window.
Password Reset Validation	Select to specify that this attribute can be used to validate the user if the user forgets his or her password.

Table C–45 (Cont.) Fields in the Editing Attribute Window

Field	Description
Searchable	By default, when a user enters a search request, the Oracle Internet Directory Self-Service Console searches based on the `cn`, `firstname`, `lastname`, and `e-mail` attributes. You can customize the attributes that can be searchable. For example, if you want to enable searching based on the attribute you are editing, then select this checkbox.The only restriction is that, to be searchable, the attribute must be cataloged.
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. When you select this type of interface, the LOV Values text area appears. In that text area, enter the values for the list, pressing the ENTER key after each one. ▪ 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

See Also: "Configuring User Entries by Using the Oracle Internet Directory Self-Service Console" on page 31-14

Table C–46 Fields in the Assign Privileges Windows

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

Table C–46 (Cont.) Fields in the Assign Privileges Windows

Privilege	Description of Access Granted
Allow privilege assignment to users	Assign access rights to users
Allow privilege assignment to groups	Assign access rights to groups
Allow service management	Enable group members to manage services for users. If this is selected, then a **Services** link becomes available in the **Directory** tab page when the latter is accessed by group members.
Allow account management	Enable group members to manage services for users. If this is selected, then an **Accounts** link becomes available in the **Directory** tab page when the latter is accessed by group members.
Allow Oracle Delegated Administration Services configuration	Configure Oracle Delegated Administration Services user interface

See Also:

- Assigning Privileges to Users by Using the Oracle Internet Directory Self-Service Console on page 31-19
- "Assigning Privileges to Groups by Using the Oracle Internet Directory Self-Service Console" on page 31-22

Identity Management Realm Fields in the Oracle Internet Directory Self-Service Console

Table C–47 Create Identity Management Realm Window for ASP Administrators

Field	Description
Basic Information	
Realm Name	Enter a relatively short version of the name of the realm for this realm. The name you enter is used to create the DN for this realm entry. This field is mandatory.
Realm Contact	Enter the name of the person to contact for any issues regarding this realm.

Table C–47 (Cont.) Create Identity Management Realm Window for ASP

Field	Description
Description	Enter any additional information about this realm. This field is optional.
Logo Management	
Enable Realm Logo	Select to display the realm logo on the Identity Management Realm Configuration window.
Enable Product Logo	Select to display the product logo on the Identity Management Realm Configuration window. **Note:** If both Enable Realm Logo and Enable Product Logo are selected, then the realm logo appears at the top, with the product logo beneath it.
Update Realm Logo	Enter the path and file name of the logo for this realm or, alternatively, navigate to it by choosing **Browse**.

Table C–48 Fields in the Identity Management Realm Window

Field	Description
Directory Configuration	
Attribute for Login Name	Enter the attribute by which you want users to identify themselves when they log in--for example, `UID`, `EmployeeNumber`, `SSN`. This is the attribute that uniquely identifies the user. Oracle Application Server Single Sign-On locates the user by using this attribute during login. When you make changes to this attribute, be sure that the user entries contain this attribute and are unique. You can enforce the uniqueness by setting up an attribute uniqueness constraint on this attribute under the user search base. This field is mandatory.
Attribute for RDN	The attribute used for creating the RDN component of the user entry. The value you enter for this field should not be the same as the value you entered in the Attribute for Login Name field.

Table C–48 (Cont.) Fields in the Identity Management Realm Window

Field	Description
User Search Base	Enter the DN of the entry under which the user entries for this realm are located.Make sure you enter the valid DN and users are present under this context. Oracle Application Server Single Sign-On looks for users under this context during user login. Also, be sure that all the ACLs are set up properly. Any discrepancy among the ACLs will disrupt either the login process or the behavior of Oracle Internet Directory Self-Service Console. This field is mandatory.

User Creation Context	Enter the DN of the entry under which to create users for this realm. This should be the same as that for the user search base.
	
	If you want to distribute the users under different contexts under the user search base, then you can set this value to be different than that of the user search base. In either case, this DN should be either that of the user search base, or of a context under the user search base. For example, if the user search base is `cn=users,dc=acme,dc=com`, and you want to divide the users based on the locality, then you can set this value to:
	
	`L=America, cn=users,dc=acme,dc=com`
	
	`L=Asia, cn=users,dc=acme,dc=com`
	
	`L=Europe, cn=users,dc=acme,dc=com`
	
	Note: The Oracle Internet Directory Self-Service Console expects these contexts to be present and the privileges under these contexts to be set correctly.

Group Search Base	Enter the DN of the entry under which group entries for this realm are located. This field is mandatory.
Group Creation Context	Enter the DN of the entry under which to create groups for this realm
Search Return Limit	Enter the maximum number to be displayed in a search. This field is mandatory.
Logo Management	
Enable Realm Logo	Select to display the realm logo on the Identity Management Realm Configuration window.

Table C–48 (Cont.) Fields in the Identity Management Realm Window

Field	Description
Enable Product Logo	Select to display the product logo on the Identity Management Realm Configuration window. **Note:** If both Enable Realm Logo and Enable Product Logo are selected, then the realm logo appears at the top, with the product logo beneath it.
Update Realm Logo	Enter the path and file name of the logo for this realm or, alternatively, navigate to it by choosing **Browse**.

See Also: "Configuring an Identity Management Realm by Using the Oracle Internet Directory Self-Service Console" on page 31-11

Resource Access Information Fields in the Oracle Internet Directory Self-Service Console

Table C–49 Fields in the Create Resource Type Window

Property	Description
Display Name	Name to be used when the resource type appears in the user interface.
Description	Textual description that explains the purpose of the resource type and any other information you want to enter for it.
Authentication Class	Leave this field blank.

Table C–49 Fields in the Create Resource Type Window

Property	Description

Connection String	Format for constructing the connection string using the values stored in Oracle Internet Directory for the resource. For example:
	■ For the Oracle9*i* Database Server or a JDBC data source your connection string format might be:
	`orclUserIDAttribute/orclPasswordAttribute @orclFlexAttribute1`
	This string indicates that the user name is followed by a slash, the password, an at sign (@), and then additional attribute 1—for example, for the TNS name of the database. A connection string that adheres to this format would look similar to this one:
	`scott/tiger@db1`
	■ For Oracle Express your connection string format might be: server=orclFlexAttribute1/domain=orclFlexAttribute2/user=orclUserIDAttribute/password=orclPasswordAttribute
	This string indicates that server= is followed by the first additional attribute, a slash, domain=, the second additional attribute, a slash, the user name, a slash, and the password. A connection string that adheres to this format would look similar to this one:
	`server=a1/domain=a2/user=scott/password=tiger`

User Name/ID Field Name	Display name of the user name field that appears on the Create Resource window when a user creates new resource access information. Typically, this display name is something like "Username" or "User Name".
Password Field Name	Display name of the password field in the Create Resource window. Typically, this display name is "Password".
Additional Fields	Display name of the additional fields displayed in the Create Resource window beyond user name and password. For example, you might use one of these fields to contain a server or domain name. Typically, this display name is descriptive of the field contents, such as "Server" or "Domain".

D

The LDAP Filter Definition

The paper contained in this appendix is copied with permission from RFC 2254 of the Internet Engineering Task Force. The URL for this document is:
`http://www.ietf.org`

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.

E

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=(LDAP_authentication_choice)|LDAP_security_choice)
LDAP_authentication_choice::= proxy | simple | MD5Digest | PKCS12
LDAP_security_choice::= SSLNoAuth | SSLOneWay | SASL

<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 earlier 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> )]+
```

F

Addition of a Directory 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 25, "Oracle Directory Replication 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.AL32UTF8
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 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;
```

This command creates 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=connect_string server=oidrepld instance=instance_number stop
$ oidctl connect=connect_string server=oidldapd instance=instance_number stop
$ oidmon connect=connect_string stop
```

In these commands, *`connect_string`* 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=connect_string server=oidrepld instance=instance_number
stop
```

Repeat this procedure on all nodes except the sponsor node. Specify appropriate net service names for the corresponding nodes.

4. Quiesce **Oracle9i Advanced 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 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

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 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 1.3.2.2-1 or later. Visit:
  `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit:
  `http://www.datafocus.com/`

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 F-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* Advanced Replication catalogs must be created. To drop the definition of the MDS from the Oracle9*i* Advanced Replication catalog on the new node and add the Oracle9*i* Advanced 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* Advanced 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 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
    - MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

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* Advanced 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.

G

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 2-13 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:** *Oracle10i 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:** *Oracle10i 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:** *Oracle10i 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.AL32UTF8`
- `JAPANESE_JAPAN.AL32UTF8`

Using Non-UTF-8 Databases

You can run the Oracle directory server and database tools on a non-UTF-8—that is, neither UTF8 nor AL31UTF8—database, but be sure that all characters in the client character set are included in the database character set (with the same or different codes). 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 UTF-8 Encoded Strings

An LDIF file Containing Only ASCII Strings

In this scenario, character strings for attribute values are also in ASCII.

Because all tools use the UTF-8 character set by default, and ASCII is a proper subset of UTF-8, all tools can interpret these files. The same is true of keyboard input of values that are simply ASCII strings.An LDIF file Containing UTF-8 Encoded Strings

An LDIF file Containing UTF-8 Encoded Strings

In this scenario, character strings for attribute values are also in UTF-8.

Because, by default, all tools use the UTF-8 character set (the Oracle character set name is AL32UTF8), 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 (the Oracle character set name is AL32UTF8) 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 `".AL32UTF8"` (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

Setting NLS_LANG in the Client Environment

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`

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 `.AL32UTF8`, 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 *Oracle10i Database Installation Guide for Windows* for more information on which character set to use for command-line tools in an MS-DOS session.

If you are using a pre-installed Oracle9*i* release 9.2 database with Oracle Internet Directory, then you must also set the database character set to UTF-8. See the *Oracle10i Database Globalization Support Guide* in the Oracle Database Documentation Library and *Oracle10i 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 Operations Command-Line 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 connect_string -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 1.3.2.2-1 or later. Visit: `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit: `http://www.datafocus.com/`

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 connect_string -b baseDN -f output_file
```

In this example, if the NLS_LANG environment variable is not set, or is set to *language_territory*`.AL32UTF8`, 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 this 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 this 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 connect_string -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 1.3.2.2-1 or later. Visit:
  `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit:
  `http://www.datafocus.com/`

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 1.3.2.2-1 or later. Visit:
  `http://sources.redhat.com`
- MKS Toolkit 6.1. Visit:
  `http://www.datafocus.com/`

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

Table H–1 standard error messages and their causes. Oracle Internet Directory also returns other messages listed and described in "Additional Error Messages" on page H-6.

Table H–1 Standard Error Messages

Error	Cause
00—LDAP_SUCCESS	The operation was successful.
01—LDAP_ OPERATIONS_ERROR	General errors encountered by the server when processing the request.

Table H–1 (Cont.) Standard Error Messages

Error	Cause
02—LDAP_PROTOCOL_ ERROR	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	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	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	Presented value is not the same as the one in the entry.
06—LDAP_COMPARE_ TRUE	Presented value is same as the one in the entry.
07—LDAP_STRONG_ AUTH_NOT_ SUPPORTED	Bind method is not supported by the server.
08—LDAP_STRONG_ AUTH_REQUIRED	Strong authentication is required. Oracle Internet Directory does not return this message at the present time.
09—LDAP_PARTIAL_ RESULTS	Server returned a referral.
10—LDAP_REFERRAL	Server returned a referral.
11—LDAP_ ADMINLIMIT_ EXCEEDED	Oracle Internet Directory does not return this message at the present time.
12—LDAP_ UNAVAILABLE_ CRITICALEXTENSION	Specified request is not supported
16—LDAP_NO_SUCH_ ATTRIBUTE	Attribute does not exist in the entry specified in the request.
17—LDAP_UNDEFINED_ TYPE	Specified attribute type is undefined in the schema.

Table H–1 (Cont.) Standard Error Messages

Error	Cause
18—LDAP_ INAPPROPRIATE_ MATCHING	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	The value in the request violated certain constraints.
20—LDAP_TYPE_OR_ VALUE_EXISTS	Duplicate values specified for the attribute.
21—LDAP_INVALID_ SYNTAX	Specified *attribute* syntax is invalid. In a search, the *filter* syntax is invalid.
32—LDAP_NO_SUCH_ OBJECT	The base specified for the operation does not exist.
33—LDAP_ALIAS_ PROBLEM	Oracle Internet Directory does not return this message at the present time.
34—LDAP_INVALID_ DN_SYNTAX	Error in the DN syntax.
35—LDAP_IS_LEAF	The entry is a leaf (terminal entry). Oracle Internet Directory does not return this message at the present time.
36—LDAP_ALIAS_ DEREF_PROBLEM	Oracle Internet Directory does not return this message at the present time.
48—LDAP_ INAPPROPRIATE_AUTH	Oracle Internet Directory does not return this message at the present time.
49—LDAP_INVALID_ CREDENTIALS	Bind failed because the credentials are not correct.
50—LDAP_ INSUFFICIENT_ACCESS	The client does not have access to perform this operation.
51—LDAP_BUSY	Server cannot accept any more client connections. Oracle Internet Directory does not return this message at the present time.
52—LDAP_ UNAVAILABLE	Cannot contact the server at all. Oracle Internet Directory does not return this message at the present time.
53—LDAP_UNWILLING_ TO_PERFORM	General error, or server is in read-only mode.
54—LDAP_LOOP_ DETECT	Oracle Internet Directory does not return this message at the present time.

Table H–1 (Cont.) Standard Error Messages

Error	Cause
64—LDAP_NAMING_ VIOLATION	Oracle Internet Directory does not return this message at the present time.
65—LDAP_OBJECT_ CLASS_VIOLATION	A change to the entry violates the objectclass definition.
66— LDAP_NOT_ ALLOWED_ON_ NONLEAF	The entry to be deleted has children.
67—LDAP_NOT_ ALLOWED_ON_RDN	Cannot perform the operation on RDN attributes—for example, you cannot delete the RDN attribute of the entry.
68—LDAP_ALREADY_ EXISTS	Duplicate ADD condition.
69—LDAP_NO_OBJECT_ CLASS_MODS	Oracle Internet Directory does not return this message at the present time.
70—LDAP_RESULTS_ TOO_LARGE	Oracle Internet Directory does not return this message at the present time.
80—LDAP_OTHER	Oracle Internet Directory does not return this message at the present time.
81—LDAP_SERVER_ DOWN	Can't contact LDAP server. This message is returned from the SDK.
82—LDAP_LOCAL_ ERROR	The client encountered an internal error. This message is returned from the client SDK.
83—LDAP_ENCODING_ ERROR	The client encountered an error in encoding the request. This message is returned from the SDK.
84—LDAP_DECODING_ ERROR	The client encountered an error in decoding the request. This message is returned from the SDK.
85—LDAP_TIMEOUT	Client encountered the time out specified for the operation. This message is returned from the SDK.
86—LDAP_AUTH_ UNKNOWN	Authentication method is unknown to the client SDK.
87—LDAP_FILTER_ ERROR	Bad search filter
88—LDAP_USER_ CANCELLED	User cancelled operation

Table H–1 (Cont.) Standard Error Messages

Error	Cause
89—LDAP_PARAM_ ERROR	Bad parameter to an LDAP routine
90—LDAP_NO_MEMORY	Out of memory

Additional Error Messages

Table H–2 lists additional error messages and their causes. These messages do not display error codes.

The Oracle Internet Directory application replaces the *parameter* tag seen in some of the following messages with the appropriate runtime value.

Table H–2 Additional Error Messages

Error	Cause
%s attribute not found	The particular attribute type is not defined in the schema.
<parameter> not found for attribute <parameter>	Value not found in the attribute. (ldapmodify)
Admin domain does not contain schema information for objectclass <parameter>	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	Duplicate object identifier specified. (schema modification)
Attribute <parameter> already in use	Duplicate attribute name. (schema modification)
Attribute <parameter> has syntax error.	Syntax error in the attribute name definition. (schema modification)
Attribute <parameter> is not supported in the schema.	Attribute not defined. (all operations)
Attribute <parameter> is single valued.	Attribute is single-valued. (ldapadd & ldapmodify)
Attribute <parameter> not present in the entry.	This attribute does not exist in the entry. (ldapmodify)
Bad attribute definition.	Syntax error in attribute definition. (schema modification)

Table H–2 (Cont.) Additional Error Messages

Error	Cause
Currently Not Supported	The version of LDAP request is not supported by this server.
Entry to be deleted not found.	DN specified in the delete operation not found.
Entry to be modified not found	The entry specified in the request is not found.
Error encountered while adding <parameter> to the entry	Returned when modify add operation is invoked. A possible cause is that the system resource is unavailable.
Error encountered while encrypting an attribute value.	Error in encrypting user password. (all operations)
Error in DN Normalization.	DN specified is invalid. Syntax error encountered in parsing the DN. (all operations)
Error in hashing <parameter> attribute.	Error in creating hash entry for the attribute. (schema modification)
Error in hashing <parameter> objectclass.	Error in creating hash entry for the objectclass. (schema modification)
Error in Schema hash creation.	Error while creating hash table for schema. (schema modification)
Error replacing <parameter>.	Error in replacing this attribute. (ldapmodify)
Error while normalizing value for attribute <parameter>.	Error in normalizing value for the attribute. (all operations)
Failed to find <parameter> in mandatory or optional attribute list.	Attribute specified does not exist in either the mandatory or optional attribute list as required by the object class(es).
Function Not Implemented	The feature/request is currently not supported.
INVALID ACI is <parameter>	The particular ACI you specified in a request is invalid.
Mandatory attribute <parameter> is not defined in Admin Domain <parameter>.	MUST refers to attribute not defined. (schema modification)

Table H–2 (Cont.) Additional Error Messages

Error	Cause
Mandatory Attribute missing.	The mandatory attribute for the particular entry is missing, as required by the particular object class.
Matching rule, <parameter>, not defined.	Matching rule not defined in the server. (schema modification)
MaxConn Reached	The maximum number of concurrent connections to the LDAP server has been reached.
Modifying the Naming attribute for the entry without modifying the DN.	Cannot modify the naming attributes using ldap_modify. A naming attribute, such as *cn* is an element in the DN.
New Parent not found.	New parent specified in modifydn operation does not exist.(ldapmodifydn)
Object already exists.	Duplicate entry. (ldapadd and ldapmodifydn)
Object ID <parameter> already in use.	Duplicate object identifier specified. (schema modification)
Objectclass <parameter> already in use.	Duplicate Objectclass name. (schema modification)
Objectclass attribute missing.	The objectclass attribute is missing for this particular entry.
OID <parameter> has syntax error.	syntax error in the object identifier definition. (schema modification)
One of the attributes in the entry has duplicate value.	You entered two values for the same attribute in the entry you are creating.
Operation not allowed on the <parameter>.	Operation not allowed on this entry. (modify, add, and delete)
Operation not allowed on the DSE Entry.	Can't do this operation on DSE entry. (delete)
Optional attribute <parameter> is not defined in Admin Domain <parameter>.	MAY refers to attribute not defined. (schema modification)
Parent entry not found in the directory.	Parent entry does not exist. (ldapadd and perhaps ldapmodifydn)

Table H–2 (Cont.) Additional Error Messages

Error	Cause
Super object <parameter> is not defined in Admin Domain <parameter>.	SUP types refer to non-existing class. (schema modification)
Super type undefined.	SUP type does not exist. (schema modification)
Super user addition not permitted.	Cannot create super user entry. (ldapadd)
Syntax, <parameter>, not defined.	Syntax not defined in the server. (schema modification)
The attribute or the value specified in the RDN does not exist in the entry.	AVA specified as the RDN does not exist in the entry. (ldapadd)
Unknown search scope	The search scope specified in the LDAP request is not recognized.
Version Not Supported	The version of the LDAP request is not supported by this server.

Password Policy Violation Error Messages

Table H–3 on page H-9 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 H–3 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.

Table H–3 (Cont.) Password Policy Violation Error Messages

Error Number	Exception	Comment or Resolution
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.

Password Policy Controls

Table H–4 Password Policy Controls

Object Identifier	Exception	Description
2.16.840.1.113894.1.8.6	`OID_PASSWORD_REQUEST_CONTROL`	The request control that the client sends to get a response from the server.
2.16.840.1.113894.1.8.7	`OID_PASSWORD_EXPWARNING_CONTROL`	The response control that the server sends when the `pwdExpireWarning` attribute is enabled and the client sends the request control. The response control value contains the time in seconds to password expiration.
2.16.840.1.113894.1.8.8	`OID_PASSWORD_GRACELOGIN_CONTROL`	The response control that the server sends when grace logins are configured and the client sends a request control. The response control value contains the remaining number of grace logins.
2.16.840.1.113894.1.8.9	`OID_PASSWORD_MUSTCHANGE_CONTROL`	The response control that the server sends when forced password reset is enabled and the client sends the request control. The client must force the user to change the password upon receipt of this control.

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 Advanced Replication**

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 Advanced 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 uniqueness

An Oracle Internet Directory feature that ensures that no two specified attributes have the same value. It enables applications synchronizing with the enterprise directory to use attributes as unique keys.

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 and Provisioning 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.

cluster

A collection of interconnected usable whole computers that is used as a single computing resource. Hardware clusters provide high availability and scalability.

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 runtime. 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 9.2 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.

default identity management realm

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 identity management realm, and the enterprises that are hosted are each associated with their own identity management realm in the DIT.

default realm location

An attribute in the root Oracle Context that identifies the root of the default identity management realm.

delegated administrator

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 an environment, a global administrator performs activities that span the entire directory. Other administrators—called delegated administrators—may exercise roles in specific identity management realms, or for specific applications.

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 profile

In an Oracle Directory Integration Platform environment, an entry in Oracle Internet Directory that describes how Oracle Directory Integration and Provisioning platform communicates with external systems and what is communicated.

directory integration and provisioning 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 and Provisioning 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 and provisioning 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 and provisioning 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. In a cold failover cluster configuration, an application running on one cluster node is transparently migrated to another cluster node. During this migration, clients accessing the service on the cluster see a momentary outage and may need to reconnect once the failover is complete.

fan-out replication

Also called a point-to-point replication, a type of replication in which a supplier replicates directly to a consumer. That consumer can then replicate to one or more other consumers. The replication can be either full or partial.

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 administrator

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 an environment, a global administrator performs activities that span the entire directory.

global unique identifier (GUID)

An identifier generated by the system and inserted into an entry when the entry is added to the directory. In a multimaster replicated environment, the GUID, not the DN, uniquely identifies an entry. The GUID of an entry cannot be modified by a user.

grace login

A login occurring within the specified period before password expiration.

group search base

In the Oracle Internet Directory default DIT, the node in the identity management realm under which all the groups can be found.

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.

identity management

The process by which the complete security lifecycle for network entities is managed in an organization. It typically refers to the management of an organization's application users, where steps in the security life cycle include account creation, suspension, privilege modification, and account deletion. The network entities managed may also include devices, processes, applications, or anything else that needs to interact in a networked environment. Entities managed by an identity management process may also include users outside of the organization, for example customers, trading partners, or Web services.

identity management realm

A collection of identities, all of which are governed by the same administrative policies. In an enterprise, all employees having access to the intranet may belong to one realm, while all external users who access the public applications of the enterprise may belong to another realm. An identity management realm is represented in the directory by a specific entry with a special object class associated with it.

identity management realm-specific Oracle Context

An Oracle Context contained in each identity management realm. It stores the following information:

- User naming policy of the identity management realm—that is, how users are named and located
- Mandatory authentication attributes

- Location of groups in the identity management realm
- Privilege assignments for the identity management realm—for example: who has privileges to add more users to the Realm.
- Application specific data for that Realm including authorizations

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**.

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.

logical host

In a cold failover cluster configuration, 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

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**

multimaster replication

Also called peer-to-peer or *n*-way replication, a type of replication that enables multiple sites, acting as equals, to manage groups of replicated data. In a multimaster replication environment, each node is both a supplier and a consumer node, and the entire directory is replicated on each node.

naming attribute

The attribute used to compose the RDN of a new user entry created through Oracle Delegated Administration Services or Oracle Internet Directory Java APIs. The default value for this is `cn`.

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.

native agent

In an Oracle Directory Integration Platform environment, an agent that runs under the control of the **directory integration and provisioning server**. It is in contrast to an **external agent**.

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

nickname attribute

The attribute used to uniquely identify a user in the entire directory. The default value for this is `uid`. Applications use this to resolve a simple user name to the complete distinguished name. The user nickname attribute cannot be multi-valued—that is, a given user cannot have multiple nicknames stored under the same attribute name.

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 enables you to create applications that use the native procedures or function calls of a third-generation language to access an Oracle database server and control all phases of SQL statement execution.

Oracle Delegated Administration Services

A set of individual, pre-defined services—called Oracle Delegated Administration Services units—for performing directory operations on behalf of a user. Oracle Internet Directory Self-Service Console makes it easier to develop and deploy administration solutions for both Oracle and third-party applications that use Oracle Internet Directory.

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 and provisioning server

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 Identity Management

An infrastructure enabling deployments to manage centrally and securely all enterprise identities and their access to various applications in the enterprise.

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 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.

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.

See Also: *Oracle Advanced Security Administrator's Guide*

Oracle9*i* Advanced Replication

A feature in Oracle9*i* that enables database tables to be kept synchronized across two Oracle databases.

other information repository

In an Oracle Directory Integration and Provisioning 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.

peer-to-peer replication

Also called multimaster replication or *n*-way replication. A type of replication that enables multiple sites, acting as equals, to manage groups of replicated data. In such

a replication environment, each node is both a supplier and a consumer node, and the entire directory is replicated on each node.

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.

point-to-point replication

Also called fan-out replication is a type of replication in which a supplier replicates directly to a consumer. That consumer can then replicate to one or more other consumers. The replication can be either full or partial.

primary node

In a cold failover cluster configuration, the cluster node on which the application runs at any given time.

See Also: **secondary node** on page Glossary-21

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.

realm search base

An attribute in the root Oracle Context that identifies the entry in the DIT that contains all identity management realms. This attribute is used when mapping a simple realm name to the corresponding entry in the directory.

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* Advanced 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.

Root Oracle Context

In the Oracle Identity Management infrastructure, the Root Oracle Context is an entry in Oracle Internet Directory containing a pointer to the default identity management realm in the infrastructure. It also contains information on how to locate an identity management realm given a simple name of the realm.

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.

secondary node

In a cold failover cluster configuration, the cluster node to which an application is moved during a failover.

See Also: **primary node** on page Glossary-18

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 these 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)**.

subACLSubentry

A specific type of subentry that contains ACL information.

subclass

An object class derived from another object class. The object class from which it is derived is called its **superclass**.

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.

subschema DN

The list of DIT areas having independent schema definitions.

subSchemaSubentry

A specific type of **subentry** containing schema information.

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.

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 for each 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 enables 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**.

UTF-16

16-bit encoding of **Unicode**.The Latin-1 characters are the first 256 code points in this standard.

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.

user search base

In the Oracle Internet Directory default DIT, the node in the identity management realm under which all the users are placed.

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 8-bit encoding of **Unicode** that uses sequences of 1, 2, 3, or 4 bytes for each character. Characters from 0-127 (the 7-bit ASCII characters) are encoded with one byte, characters from 128-2047 require two bytes, characters from 2048-65535 require three bytes, and characters beyond 65535 require four bytes. The Oracle character set name for this is AL32UTF8 (for the Unicode 3.1 standard).

virtual host name

In a cold failover cluster configuration, the host name corresponding to this virtual IP address.

virtual IP address

In a cold failover cluster configuration, each physical node has its own physical IP address and physical host name. To present a single system image to the outside world, the cluster uses a dynamic IP address that can be moved to any physical node in the cluster. This is called the virtual IP address.

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