

Oracle® Internet Directory

Administrator's Guide,
10g Release 2 (10.1.2)

B14082-02

July 2005

Oracle Internet Directory Administrator's Guide, 10g Release 2 (10.1.2)

B14082-02

Copyright © 1999, 2005, Oracle. All rights reserved.

Primary Authors: Ellen Desmond, Richard Smith

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

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

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

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software—Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

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

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.



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

Oracle Directory Manager requires the Java™ Runtime Environment. The Java™ Runtime Environment, Version

JRE 1.1.6. ("The Software") is developed by Sun Microsystems, Inc. 2550 Garcia Avenue, Mountain View, California 94043. Copyright (c) 1997 Sun Microsystems, Inc.

This product contains SSLPlus Integration Suite™ version 1.2, from Consensus Development Corporation.

iPlanet is a registered trademark of Sun Microsystems, Inc.

Contents

Preface	xxix
Audience	xxix
Documentation Accessibility	xxix
Related Documents	xxx
Conventions	xxx
What's New in Oracle Internet Directory?	xxxiii
New Features Introduced with Oracle Internet Directory 10g Release 2 (10.1.2)	xxxiii
New Features Introduced with Oracle Internet Directory 10g (9.0.4)	xxxiv
About Oracle Internet Directory Release 9.2	xxxix
New Features Introduced with Oracle Internet Directory Release 9.0.2	xl
New Features Introduced with Oracle Internet Directory Release 3.0.1	xliv
New Features Introduced with Oracle Internet Directory Release 2.1.1	xlvi
Part I Getting Started	
1 Introduction to LDAP and Oracle Internet Directory	
What Is a Directory?	1-1
The Expanding Role of Online Directories	1-1
The Problem: Too Many Special-Purpose Directories	1-2
What Is the Lightweight Directory Access Protocol (LDAP)?	1-3
LDAP and Simplified Directory Management	1-3
LDAP Version 3	1-3
Oracle Identity Management	1-4
What Is Oracle Internet Directory?	1-5
Overview of Oracle Internet Directory	1-5
Components of Oracle Internet Directory	1-6
Advantages of Oracle Internet Directory	1-7
Scalability	1-7
High Availability	1-7
Security	1-7
Integration with the Oracle Environment	1-7
How Oracle Components Use Oracle Internet Directory	1-7
Easier and More Cost-Effective Administration of Applications	1-8
Tighter Security Through Centralized Security Policy Administration	1-8

Integration of Multiple Directories.....	1-9
--	-----

2 Directory Concepts and Architecture

Oracle Internet Directory Architecture	2-1
An Oracle Internet Directory Node.....	2-2
An Oracle Directory Server Instance.....	2-3
Directory Metadata	2-4
Configuration Set Entries.....	2-6
Example: How Oracle Internet Directory Works	2-6
Entries	2-7
Distinguished Names (DNs) and Directory Information Trees (DITs).....	2-7
Entry Caching	2-8
Attributes	2-8
Kinds of Attribute Information	2-9
Single-Valued and Multivalued Attributes	2-10
Common LDAP Attributes	2-10
Attribute Syntax	2-11
Attribute Matching Rules.....	2-11
Attribute Options	2-11
Object Classes	2-12
Subclasses, Superclasses, and Inheritance.....	2-12
Object Class Types	2-12
Structural Object Classes.....	2-13
Auxiliary Object Classes	2-13
Abstract Object Classes	2-13
Naming Contexts	2-14
Security	2-14
Globalization Support	2-15
Distributed Directories	2-16
Directory Replication.....	2-17
Directory Partitioning.....	2-18
Knowledge References and Referrals	2-19
Oracle Delegated Administration Services and the Oracle Internet Directory Self-Service Console	2-21
The Service Registry and Service to Service Authentication	2-21
Oracle Directory Integration and Provisioning	2-22
Oracle Internet Directory and Identity Management	2-22
About Identity Management	2-23
About the Oracle Identity Management Infrastructure	2-23
Identity Management Realms	2-25
Default Identity Management Realm.....	2-25
Identity Management Policies.....	2-25
Resource Information	2-26
Resource Type Information	2-26
Resource Access Information	2-26
Location of Resource Information in the DIT	2-27

3 Post-Installation Tasks and Information

Task 1: Reset the Default Security Configuration	3-1
Task 2: Reset the Default Password for the Database	3-2
Task 3: Run the OID Database Statistics Collection Tool	3-3
Tasks to Perform After Upgrading from Release 9.0.2	3-3
Set ACL Policy on Groups Container after Upgrade from Release 9.0.2.....	3-3
Determining LDAP Port Assignment on UNIX and Linux	3-4
Log File Locations	3-4

4 Directory Administration Tools

Using Oracle Directory Manager	4-1
Starting Oracle Directory Manager	4-2
Connecting to a Directory Server by Using Oracle Directory Manager	4-2
Navigating Oracle Directory Manager	4-3
Overview of Oracle Directory Manager	4-3
The Oracle Directory Manager Menu Bar	4-3
The Oracle Directory Manager Toolbar.....	4-5
Connecting to Additional Directory Servers by Using Oracle Directory Manager	4-5
Disconnecting from a Directory Server by Using Oracle Directory Manager	4-6
Configuring the Display and Duration of Searches in Oracle Directory Manager	4-6
Performing Administrative Tasks by Using Oracle Directory Manager	4-7
Process Control of Oracle Internet Directory Components	4-8
Tools and Daemons Important to Oracle Internet Directory Process Control.....	4-8
Oracle Internet Directory Integration with OPMN.....	4-8
Semantics of OPMN Monitoring Oracle Internet Directory.....	4-9
Oracle Internet Directory Snippet in OPMN.XML	4-9
Semantics of OPMN Starting Oracle Internet Directory	4-9
Semantics of OPMN Stopping Oracle Internet Directory	4-10
Semantics of OPMN Monitoring OIDMON	4-10
Oracle Internet Directory Process Control–Best Practices	4-10
Changing the Configuration of the Default OID LDAP Server Instance	4-11
Configuring Additional Oracle Internet Directory LDAP Server Instances	4-12
Deconfiguring the Default Oracle Internet Directory LDAP Server Instance	4-12
Configuring an Instance of the Oracle Internet Directory Replication Server.....	4-12
Configuring an Oracle Directory Integration and Provisioning Server Instance.....	4-13
OIDMON and the ODS_PROCESS Table.....	4-13
OIDCTL Process Control Semantics.....	4-14
Using Command-Line Tools	4-14
Command-Line Tools for Starting, Stopping, and Monitoring Oracle Internet Directory Servers.....	4-15
Command-Line Tools for Managing Entries and Attributes.....	4-16
Command-Line Tools for Performing Bulk Operations.....	4-17
Command-Line Tools for Managing Replication.....	4-17
OID Migration Tool (ldifmigrator)	4-18
OID Database Statistics Tool (oidstats.sql).....	4-18
OID Database Password Utility (oidpasswd)	4-19

Routine Administration at a Glance	4-19
---	------

Part II Basic Directory Administration

5 Oracle Directory Server Administration

Managing Server Configuration Set Entries	5-1
Preliminary Considerations for Managing Configuration Set Entries.....	5-2
Managing Server Configuration Set Entries by Using Oracle Directory Manager	5-3
Viewing Configuration Set Entries by Using Oracle Directory Manager	5-3
Adding Configuration Set Entries by Using Oracle Directory Manager.....	5-4
Modifying Configuration Set Entries by Using Oracle Directory Manager.....	5-5
Deleting Configuration Set Entries by Using Oracle Directory Manager	5-5
Managing Server Configuration Set Entries by Using Command-Line Tools	5-5
Adding Configuration Set Entries by Using ldapadd.....	5-6
Modifying and Deleting Configuration Set Entries by Using ldapmodify	5-6
Setting System Operational Attributes	5-7
Setting System Operational Attributes by Using Oracle Directory Manager	5-7
Setting System Operational Attributes by Using ldapmodify.....	5-7
Managing Naming Contexts	5-8
Publishing Naming Contexts by Using Oracle Directory Manager	5-8
Publishing Naming Contexts by Using ldapmodify.....	5-8
Managing Super Users, Guest Users, and Proxy Users	5-8
About Super Users, Guest Users, and Proxy Users	5-9
Managing Super Users, Guest Users, and Proxy Users by Using Oracle Directory Manager	5-9
Managing Super Users, Guest Users, and Proxy Users by Using ldapmodify.....	5-10
Managing Anonymous Binds	5-10
Viewing Active Server Instance Information	5-11
Closing Idle LDAP Connections	5-11
Changing the Password to the Oracle Internet Directory Database Server	5-11
Dereferencing Alias Entries	5-11
About Alias Entries.....	5-12
Examples: Using Alias Entry Dereferencing.....	5-12
Example: Adding an Alias Entry	5-13
Examples: Searching the Directory with Alias Entries.....	5-14
Example: Modifying Alias Entries	5-15
Success and Error Messages	5-15
Locating Directory Servers in a Distributed Environment	5-16
Static Directory Server Discovery by Using the Directory Server Usage File (ldap.ora)	5-16
Dynamic Directory Server Discovery by Using the Domain Name System (DNS)	5-17
How a Client Locates a Directory Server by Using DNS.....	5-17
Registering a Directory Server with the Domain Name System.....	5-18

6 Directory Entries Administration

Managing Entries by Using Oracle Directory Manager	6-1
Searching for Entries by Using Oracle Directory Manager.....	6-1
Viewing Attributes for a Specific Entry by Using Oracle Directory Manager	6-3

Adding Entries by Using Oracle Directory Manager	6-3
Adding a New Entry by Using Oracle Directory Manager	6-3
Adding an Entry by Copying an Existing Entry in Oracle Directory Manager	6-4
Example: Adding a User Entry by Using Oracle Directory Manager	6-4
Modifying Entries by Using Oracle Directory Manager	6-5
Example: Modifying a User Entry by Using Oracle Directory Manager	6-5
Managing Entries with Attribute Options by Using Oracle Directory Manager	6-6
Adding an Attribute Option to an Existing Entry by Using Oracle Directory Manager ..	6-6
Modifying an Attribute Option by Using Oracle Directory Manager	6-6
Deleting an Attribute Option by Using Oracle Directory Manager	6-7
Managing Entries by Using Command-Line Tools	6-7
Command-Line Tools for Managing Entries	6-7
Example: Adding a User Entry by Using ldapadd	6-8
Example: Modifying a User Entry by Using ldapmodify	6-8
Managing Entries with Attribute Options by Using Command-Line Tools	6-9
Example: Adding an Attribute Option by Using ldapmodify	6-9
Example: Deleting an Attribute Option by Using ldapmodify	6-9
Example: Searching for Entries with Attribute Options by Using ldapsearch	6-9
Managing Entries by Using Bulk Tools	6-9
Importing an LDIF File by Using bulkload	6-10
Task 1: Back Up the Oracle Database Server	6-10
Task 2: Find Out the Oracle Internet Directory Password	6-11
Task 3: Check Input for Schema and Data Consistency Violations	6-11
Task 4: Generate the Input Files for SQL*Loader	6-11
Task 5: Load the Input Files	6-11
If Bulk Loading Fails	6-11
Converting Directory Data to LDIF	6-12
Modifying a Large Number of Entries	6-12
Deleting a Large Number of Entries	6-12
Managing Knowledge References and Referrals	6-12
Configuring Smart Referrals	6-12
Configuring Default Referrals	6-13
Client-Side Referral Caching	6-14
How Client-Side Referral Caching Works	6-14

7 Attribute Uniqueness in the Directory

About Attribute Uniqueness	7-1
Rules for Creating Attribute Uniqueness	7-2
Specifying Multiple Attribute Names in an Attribute Uniqueness Constraint	7-3
Specifying Multiple Subtrees in an Attribute Uniqueness Constraint	7-3
Specifying Multiple Scopes in an Attribute Uniqueness Constraint	7-4
Specifying Multiple Object Classes in an Attribute Uniqueness Constraint	7-4
Specifying Multiple Subtrees, Scopes, and Object Classes in an Attribute Uniqueness Constraint	7-5
Managing Attribute Uniqueness	7-5
Location of Attribute Uniqueness Entries	7-5
Managing Attribute Uniqueness by Using Oracle Directory Manager	7-5

Creating an Attribute Uniqueness Constraint Entry	7-5
Modifying an Attribute Uniqueness Constraint Entry by Using Oracle Directory Manager.....	7-6
Deleting an Attribute Uniqueness Constraint Policy by Using Oracle Directory Manager.....	7-6
Managing Attribute Uniqueness by Using Command-Line Tools.....	7-6
Enabling and Disabling Attribute Uniqueness by Using Command-Line Tools	7-7
Creating Attribute Uniqueness Constraint Entries by Using Command-Line Tools	7-7
Modifying Attribute Uniqueness Constraint Entries by Using Command-Line Tools....	7-8
Deleting Attribute Uniqueness Constraint Entries by Using Command-Line Tools	7-8
Limitations of Attribute Uniqueness in Oracle Internet Directory 10g Release 2 (10.1.2)	7-8

8 Directory Schema Administration

About the Directory Schema	8-1
Object Classes in the Directory	8-1
About Object Class Management	8-2
Inheritance	8-2
Mandatory and Optional Attributes in Object Classes	8-2
Addition of Entries in Top-Down Sequence	8-2
Object Class Explosion	8-3
Guidelines for Adding, Modifying, and Deleting Object Classes.....	8-3
Guidelines for Adding Object Classes	8-3
Guidelines for Modifying Object Classes	8-3
Guidelines for Deleting Object Classes.....	8-4
Managing Object Classes by Using Oracle Directory Manager	8-4
Searching for Object Classes by Using Oracle Directory Manager	8-4
Viewing Properties of Object Classes by Using Oracle Directory Manager	8-5
Adding Object Classes by Using Oracle Directory Manager	8-5
Modifying Object Classes by Using Oracle Directory Manager	8-6
Deleting Object Classes by Using Oracle Directory Manager	8-6
Managing Object Classes by Using Command-Line Tools.....	8-7
Example: Adding a New Object Class	8-7
Example: Adding a New Attribute to an Auxiliary or User-Defined Object Class	8-7
Attributes in the Directory	8-8
About Attribute Management.....	8-8
Rules for Adding Attributes.....	8-8
Rules for Modifying Attributes.....	8-9
Rules for Deleting Attributes	8-9
Managing Attributes by Using Oracle Directory Manager	8-9
Viewing All Directory Attributes by Using Oracle Directory Manager.....	8-9
Searching for Attributes by Using Oracle Directory Manager.....	8-9
Adding an Attribute by Using Oracle Directory Manager	8-10
Modifying an Attribute by Using Oracle Directory Manager.....	8-11
Deleting an Attribute by Using Oracle Directory Manager.....	8-11
Indexing an Attribute by Using Oracle Directory Manager	8-12
Managing Attributes by Using Command-Line Tools.....	8-13
Adding and Modifying Attributes by Using ldapmodify	8-13

Deleting Attributes by Using ldapmodify	8-13
Indexing an Attribute by Using Command-Line Tools	8-14
How to Extend the Number of Attributes Associated with Entries	8-15
Extending the Number of Attributes Prior to Creating Entries in the Directory	8-15
Extending the Number of Attributes for Existing Entries by Creating an Auxiliary Object Class	8-16
Extending the Number of Attributes for Existing Entries by Creating a Content Rule	8-16
Rules for Creating and Modifying Content Rules	8-16
Schema Enforcement When Using Content Rules	8-17
Searches for Object Classes Listed in Content Rules	8-17
Managing Content Rules	8-18
Matching Rules in the Directory	8-19
Viewing Matching Rules by Using Oracle Directory Manager	8-19
Viewing Matching Rules by Using ldapsearch	8-20
Syntaxes in the Directory	8-20
Viewing Syntaxes by Using Oracle Directory Manager	8-20
Viewing Syntaxes by Using by Using ldapsearch	8-20

9 Dynamic and Static Groups in Oracle Internet Directory

About Groups	9-1
Static Groups	9-1
Schema Elements for Creating Static Groups	9-1
Dynamic Groups	9-2
Schema Elements for Creating a Dynamic Group	9-2
Limitations of Dynamic Groups in Oracle Internet Directory 10g Release 2 (10.1.2)	9-4
Hierarchies	9-4
Querying Group Entries	9-5
When to Use Each Kind of Group	9-5
Managing Group Entries	9-5
Managing Static Group Entries by Using Oracle Directory Manager	9-6
Creating Static Group Entries by Using Oracle Directory Manager	9-6
Modifying a Static Group Entry by Using Oracle Directory Manager	9-6
Managing Static Group Entries by Using Command-Line Tools	9-7
Creating a Static Group Entry by Using ldapadd	9-7
Modifying a Static Group by Using ldapmodify	9-7
Examples of Dynamic Group Entries	9-8
Example: a Dynamic Group Entry Using the labeledURI Attribute	9-8
Example: a Dynamic Group Entry Using the CONNECTBY Assertion	9-8
Managing Dynamic Groups by Using Oracle Directory Manager	9-9
Creating Dynamic Group Entries by Using Oracle Directory Manager	9-9
Modifying a Dynamic Group Entry by Using Oracle Directory Manager	9-10
Managing Dynamic Groups by Using Command-Line Tools	9-10
Creating a Dynamic Group Entry by Using ldapadd	9-10
Example: Creating a Dynamic Group Entry by Using ldapadd	9-11
Example: Modifying a Dynamic Group by Using ldapmodify	9-11

10 Logging, Auditing, and Monitoring the Directory

Using Debug Logging	10-1
About Oracle Internet Directory Debug Logging	10-1
About Log Messages.....	10-1
Log Messages for Specified LDAP Operations.....	10-2
Log Messages Not Associated with Specified LDAP Operations	10-2
Example: Trace Messages in Oracle Internet Directory Server Log File.....	10-2
How to Interpret Trace Messages in the Log File	10-3
Setting Debug Logging Levels	10-4
Setting Debug Logging Levels by Using Oracle Directory Manager	10-4
Setting Debug Logging Levels by Using the OID Control Utility	10-4
Setting the Operation Debug Dimension	10-5
Setting the Operation Debug Dimension by Using Oracle Directory Manager	10-6
Setting the Operation Debug Dimension by Using ldapmodify	10-6
Force Flushing the Trace Information to a Log File	10-6
Using the Audit Log	10-7
Structure of Audit Log Entries	10-8
Position of Audit Log Entries in the DIT	10-9
Auditable Events.....	10-9
Setting the Audit Level.....	10-10
Setting the Audit Level by Using Oracle Directory Manager	10-10
Setting the Audit Level by Using ldapmodify.....	10-11
Searching for Audit Log Entries	10-11
Searching for Audit Log Entries by Using Oracle Directory Manager	10-11
Searching for Audit Log Entries by Using ldapsearch.....	10-12
Purging the Audit Log	10-12
Monitoring Oracle Internet Directory Servers	10-12
Capabilities of Oracle Internet Directory Server Manageability	10-13
Oracle Internet Directory Server Manageability Architecture and Components.....	10-14
Location of Configuration Information for Oracle Internet Directory Server Manageability	10-15
Configuring Oracle Internet Directory Server Manageability.....	10-15
Configuring Critical Events.....	10-16
Using the Oracle Internet Directory Server Manageability Framework Through Oracle Enterprise Manager 10g Application Server Control Console	10-17
Enabling Information Collection by Using Oracle Enterprise Manager 10g Application Server Control Console	10-17
Starting a New Directory Server Instance by Using Oracle Enterprise Manager 10g Application Server Control Console	10-17
Stopping a Directory Server Instance by Using Oracle Enterprise Manager 10g Application Server Control Console	10-18
Restarting a Directory Server Instance by Using Oracle Enterprise Manager 10g Application Server Control Console	10-18
Viewing Directory Server Activities by Using Oracle Enterprise Manager 10g Application Server Control Console	10-18
Viewing Directory Server Operations by Using Oracle Enterprise Manager 10g Application Server Control Console	10-19

11	Backup and Restoration of a Directory	
	Backing Up and Restoring a Small Directory or Specific Naming Context	11-1
	Backing Up and Restoring a Large Directory	11-1
12	Directory Security Concepts	
	Data Integrity and Oracle Internet Directory	12-1
	Data Privacy and Oracle Internet Directory	12-1
	Authorization in Oracle Internet Directory	12-2
	Authentication in Oracle Internet Directory	12-3
	Direct Authentication	12-3
	Indirect Authentication	12-4
	External Authentication	12-6
	Protection of User Passwords for Directory Authentication	12-6
	Password Policies in Oracle Internet Directory	12-6
	Authentication by Using Simple Authentication and Security Layer (SASL)	12-7
13	Secure Sockets Layer (SSL) and the Directory	
	Supported Cipher Suites	13-1
	SSL Client Scenarios	13-2
	Limitations of the Use of SSL in 10g Release 2 (10.1.2)	13-2
	Configuring and Testing Oracle Internet Directory With SSL	13-2
	Configuring SSL Parameters	13-2
	Configuring SSL Parameters by Using Oracle Directory Manager	13-3
	Configuring SSL Parameters by Using Command-Line Tools	13-4
	Configure Oracle Internet Directory for SSL	13-4
	Testing SSL Connections From the Command Line	13-7
	Testing SSL With Encryption Only	13-7
	Testing SSL With Server Authentication	13-7
	Testing SSL With Client and Server Authentication	13-7
	Testing SSL Connections With Oracle Directory Manager	13-8
	Other Components and SSL	13-8
14	Directory Access Control	
	Overview of Access Control Policy Administration	14-1
	Access Control Management Constructs	14-2
	Access Control Policy Points (ACPs)	14-2
	The orclACI Attribute for Prescriptive Access Control	14-2
	The orclEntryLevelACI Attribute for Entry-Level Access Control	14-2
	Security Groups	14-3
	Access Control Information Components	14-6
	Object: To What Are You Granting Access?	14-6
	Subject: To Whom Are You Granting Access?	14-7
	Operations: What Access Are You Granting?	14-8
	Access Level Requirements for LDAP Operations	14-10
	How ACL Evaluation Works	14-10

Precedence Rules Used in ACL Evaluation.....	14-11
Precedence at the Entry Level.....	14-11
Precedence at the Attribute Level.....	14-12
Use of More Than One ACI for the Same Object.....	14-12
Exclusionary Access to Directory Objects	14-13
ACL Evaluation For Groups.....	14-13
Managing Access Control by Using Oracle Directory Manager.....	14-13
Configuring Oracle Directory Manager for Access Control Management.....	14-14
Configuring the Display of ACPs in Oracle Directory Manager	14-14
Configuring Searches for ACPs When Using Oracle Directory Manager.....	14-14
Viewing an ACP by Using Oracle Directory Manager.....	14-15
Adding an ACP by Using Oracle Directory Manager.....	14-15
Task 1: Specify the Entry That Will Be the ACP.....	14-16
Task 2: Configure Structural Access Items.....	14-16
Task 3: Configure Content Access Items	14-17
Adding an ACP by Using the ACP Creation Wizard of Oracle Directory Manager	14-18
Task 1: Specify the Entry That Will Be the ACP	14-18
Task 2: Configure Structural Access Items by Using the ACP Creation Wizard	14-19
Task 3: Configure Content Access Items by Using the ACP Creation Wizard.....	14-20
Modifying an ACP by Using Oracle Directory Manager.....	14-21
Task 1: Specify the Entry That You Want to Modify	14-21
Task 2: Modify Structural Access Items.....	14-21
Task 3: Modify Content Access Items	14-23
Granting Entry-Level Access by Using Oracle Directory Manager.....	14-24
Example: Managing ACPs by Using Oracle Directory Manager.....	14-24
Create a New ACP.....	14-24
Create a Third ACL.....	14-31
Create a Fourth ACL.....	14-33
Managing Access Control by Using Command-Line Tools.....	14-34
Example: Restricting the Kind of Entry a User Can Add.....	14-35
Example: Setting Up an Inheritable ACP by Using ldapmodify	14-36
Example: Setting Up Entry-Level ACIs by Using ldapmodify.....	14-36
Example: Using Wild Cards	14-36
Example: Selecting Entries by DN.....	14-37
Example: Using Attribute and Subject Selectors	14-37
Example: Granting Read-Only Access	14-38
Example: Granting Selfwrite Access to Group Entries	14-38
Example: Defining a Completely Autonomous Policy to Inhibit Overriding Policies	14-38

Part III Directory Security

15 Password Policies in Oracle Internet Directory

About Password Policies.....	15-1
What a Password Policy Is.....	15-1
Default Password Policy	15-1
Directory Server Verification of Password Policy Information.....	15-3
Overview: Establishing a Password Policy for an Identity Management Realm	15-4

Managing Password Policies	15-4
Managing Password Policies by Using Oracle Directory Manager.....	15-5
Viewing Password Policies of an Identity Management Realm by Using Oracle Directory Manager.....	15-5
Modifying Password Policies of an Identity Management Realm by Using Oracle Directory Manager.....	15-6
Managing Password Policies by Using Command-Line Tools	15-6
Example: Setting Password Policies by Using Command-Line Tools	15-6
Examples: Managing the Password Policies of an Identity Management Realm by Using Command-Line Tools	15-7
Example: Enabling and Disabling Accounts by Using Command-Line Tools.....	15-7
Example: Unlocking Accounts by Using Command-Line Tools	15-7
Example: Forcing a Password Change by Using Command-Line Tools.....	15-8
Managing Password Policies by Using the Self-Service Console	15-8
Enabling and Disabling Accounts by Using the Oracle Internet Directory Self-Service Console	15-8
Unlocking Accounts by Using the Oracle Internet Directory Self-Service Console.....	15-8
Resetting Your Own Password by Using the Oracle Internet Directory Self-Service Console	15-8
Password Policy Error Messages	15-9

16 Directory Storage of Password Verifiers

About Centralized Storage of User Authentication Credentials	16-1
Storing and Managing Password Verifiers for Authenticating to Oracle Internet Directory	16-1
Password Verifiers and Authentication to the Directory	16-2
Hashing Schemes for Creating Password Verifiers	16-2
Managing Password Protection by Using Oracle Directory Manager	16-3
Managing Password Protection by Using ldapmodify	16-3
Storing and Managing Password Verifiers for Authenticating to Oracle Components.....	16-4
About Password Verifiers for Oracle Components	16-4
Attributes for Storing Password Verifiers	16-5
Default Verifiers for Oracle Components.....	16-7
Example: How Password Verification Works for an Oracle Component	16-8
Managing Password Verifier Profiles for Oracle Components by Using Oracle Directory Manager.....	16-9
Viewing and Modifying a Password Verifier Profile for an Oracle Component by Using Oracle Directory Manager	16-10
Managing Password Verifier Profiles for Oracle Components by Using Command-Line Tools	16-10
Viewing a Password Verifier Profile by Using Command-Line Tools	16-10
Example: Modifying a Password Verifier Profile by Using Command-Line Tools.....	16-10
Verifier Generation Using Dynamic Parameters	16-10
Generating Dynamic Password Verifiers	16-11
Configuring Oracle Internet Directory to Generate Dynamic Password Verifiers	16-11

17 Delegation of Privileges for an Oracle Technology Deployment

Delegation in the Oracle Identity Management Model	17-1
---	-------------

How Delegation Works.....	17-1
Delegation in an Oracle Application Server Environment	17-2
About the Default Configuration.....	17-3
Overview: Privileges for Administering the Oracle Technology Stack	17-3
Delegation of Privileges for User and Group Management	17-4
How Privileges Are Granted for Managing User and Group Data	17-4
Default Privileges for Managing User Data	17-5
Creating Users for a Realm.....	17-5
Modifying Attributes of a User	17-5
Deleting a User	17-6
Delegating User Administration.....	17-6
Default Privileges for Managing Group Data.....	17-7
Creating Groups.....	17-7
Modifying the Attributes of Groups	17-7
Deleting Groups	17-7
Delegating Group Administration	17-8
Delegation of Privileges for Deployment of Oracle Components	17-8
How Deployment Privileges Are Granted	17-9
Oracle Application Server Administrators.....	17-9
User Management Application Administrators.....	17-10
Trusted Application Administrators.....	17-10
Delegation of Privileges for Component Runtime	17-10
Default Privileges for Reading and Modifying User Passwords	17-11
Default Privileges for Comparing User Passwords	17-12
Default Privileges for Comparing Password Verifiers	17-12
Default Privileges for Proxying on Behalf of End Users	17-13
Default Privileges for Managing the Oracle Context.....	17-13
Default Privileges for Reading Common User Attributes	17-13
Default Privileges for Reading Common Group Attributes.....	17-14
Default Privileges for Reading the Service Registry	17-14
Default Privileges for Administering the Service Registry.....	17-14

Part IV Directory Deployment

18 Directory Deployment Considerations

The Expanding Role of Directories	18-1
Logical Organization Of Directory Information	18-2
Physical Distribution: Partitions, Replicas, and High Availability	18-2
An Ideal Deployment	18-2
Partitioning Considerations.....	18-3
Replication Considerations.....	18-3
High Availability Considerations.....	18-4
Oracle Directory Integration and Provisioning	18-5
Capacity Planning, Sizing, and Tuning	18-5
Capacity Planning	18-6
Sizing Considerations.....	18-6
Tuning Considerations.....	18-8

19 Deployment of Oracle Identity Management Realms

Planning the Directory Information Tree for Identity Management	19-1
Planning the Overall Directory Structure.....	19-3
Planning the Names and Containment of Users and Groups	19-3
Considerations for Users.....	19-4
Considerations for Groups	19-4
Planning the Identity Management Realm.....	19-5
Migrating a DIT from a Third-Party Directory.....	19-7
Identity Management Realms in an Enterprise Deployment	19-7
Single Identity Management Realm in the Enterprise.....	19-7
Multiple Identity Management Realms in the Enterprise.....	19-8
Identity Management Realms in a Hosted Deployment	19-8
Identity Management Realm Implementation in Oracle Internet Directory	19-9
Default Directory Information Tree and the Identity Management Realm	19-10
Administration of Identity Management Realms	19-12
Customizing the Default Identity Management Realm	19-12
Changing the Location of Users and Groups In The Default Identity Management Realm	19-13
Creating Additional Identity Management Realms for Hosted Deployments.....	19-18

20 Capacity Planning for the Directory

About Capacity Planning	20-1
Getting to Know Directory Usage Patterns: A Case Study	20-2
I/O Subsystem Requirements	20-4
About the I/O Subsystem.....	20-5
Rough Estimates of Disk Space Requirements	20-5
Detailed Calculations of Disk Space Requirements	20-6
Memory Requirements	20-9
Network Requirements	20-10
CPU Requirements	20-11
CPU Configuration	20-11
Rough Estimates of CPU Requirements	20-11
Detailed Calculations of CPU Requirements	20-12
Summary of Capacity Plan for Acme Corporation	20-13

21 Tuning Considerations for the Directory

About Tuning	21-1
Tools for Performance Tuning	21-2
CPU Usage Tuning	21-3
Tuning CPU for Oracle Internet Directory Processes	21-3
Tuning CPU for Oracle Foreground Processes.....	21-4
Taking Advantage of Processor Affinity on SMP Systems.....	21-5
Other Alternatives for a CPU Constrained System.....	21-5
Memory Tuning	21-5
Tuning the System Global Area (SGA) for the Oracle Database.....	21-5
Other Alternatives for a Memory-Constrained System	21-6

Disk Tuning	21-6
Database Tuning	21-6
Required Parameters	21-7
Parameters Dependent on Oracle Internet Directory Server Configuration.....	21-7
Using Shared Server Process	21-8
SGA Parameters Dependent on Hardware Resources	21-8
Entry Caching	21-8
Optimizing Searches	21-9
Optimizing Searches for Large Group Entries.....	21-9
Optimizing Searches for Skewed Attributes.....	21-9
Optimizing Searches for Skewed Attributes by Using Oracle Directory Manager.....	21-9
Optimizing Searches for Skewed Attributes by Using ldapmodify.....	21-10
Setting the Time Limit Mode	21-10
Setting the Time Limit Mode by Using Oracle Directory Manager.....	21-10
Setting the Time Limit Mode by Using ldapmodify	21-10
Setting the Timeout for Client/Server Connections	21-10

22 Garbage Collection in Oracle Internet Directory

About the Oracle Internet Directory Garbage Collection Framework	22-1
Components of the Oracle Internet Directory Garbage Collection Framework.....	22-1
Garbage Collection Plug-in	22-1
Background Database Processes.....	22-2
How Oracle Internet Directory Garbage Collection Works.....	22-3
Garbage Collector Entries and the Oracle Internet Directory Statistics Collector Entry	22-4
Change Log Purging in Multimaster Replication	22-5
Modifying Oracle Internet Directory Garbage Collectors	22-6
Modifying a Garbage Collector by Using Oracle Directory Manager.....	22-6
Modifying a Garbage Collector by Using Command-Line Tools.....	22-6
Example 1: Modifying a Garbage Collector	22-6
Example 2: Disabling a Garbage Collector Change Log	22-7
Modifying the Oracle Internet Directory Statistics Collector	22-7
Enabling and Disabling Logging for Oracle Internet Directory Garbage Collectors	22-7
Enabling Logging for Oracle Internet Directory Garbage Collectors.....	22-7
Disabling Logging for Oracle Internet Directory Garbage Collectors.....	22-8
Monitoring Garbage Collection Logging.....	22-8

23 Migration of Data from Other Data Repositories

The Default Directory Structure of Oracle Internet Directory	23-1
Migrating Data from LDAP-Compliant Directories	23-2
Tools	23-2
bulkload.....	23-2
dipassistant	23-2
Oracle Directory Integration and Provisioning Server.....	23-3
Common Usage Scenarios	23-3
Scenario 1: Using an LDIF File and bulkload	23-3
Scenario 2: Using dipassistant Directly	23-4
Scenario 3: Using an LDIF File and dipassistant	23-4

Scenario 4: Using dipassistant, bulkload, and LDIF Files.....	23-5
Scenario 5: Using the Oracle Directory Integration and Provisioning Server	23-6
Tasks For Migrating Data from LDAP-Compliant Directories	23-6
Task 1: Export Data from the Non-Oracle Internet Directory Server into LDIF File Format.....	23-6
Task 2: Analyze the LDIF User Data for Any Required Schema Additions Referenced in the LDIF Data	23-6
Task 3: Extend the Schema in Oracle Internet Directory	23-7
Task 4: Remove Any Proprietary Directory Data from the LDIF File.....	23-7
Task 5: Remove Operational Attributes from the LDIF File.....	23-7
Task 6: Remove Incompatible userPassword Attribute Values from the LDIF File	23-7
Task 7: Run the bulkload.sh -check Mode and Determine Any Remaining Schema Violations or Duplication Errors.....	23-8
Migrating User Data from Application-Specific Repositories.....	23-8
The Intermediate Template File	23-8
Reconciling Data in Application Repository with Data Already in Oracle Internet Directory	23-9
Tasks For Migrating Data from Application-Specific Repositories.....	23-9
Task 1: Create an Intermediate Template File	23-9
Task 2: Run the OID Migration Tool.....	23-11

Part V Directory Replication

24 Oracle Internet Directory Replication Concepts

About Directory Replication.....	24-1
Full and Partial Directory Replication	24-2
Full Directory Replication.....	24-2
Partial Directory Replication	24-2
Directory Replication Groups.....	24-3
Data Transfer Between Nodes in a Directory Replication Group.....	24-3
Single-Master Replication Groups.....	24-4
Multimaster Replication Groups	24-4
Fan-Out Replication Groups.....	24-5
Types of Directory Replication Compared.....	24-6
Multimaster Replication with Fan-Out.....	24-6
Included and Excluded Naming Contexts.....	24-7
Replication Agreements.....	24-8
Oracle Database Advanced Replication Agreements	24-8
LDAP-Based Replication Agreements	24-8
Replication Configuration Objects in the Directory	24-9
The Replication Configuration Container	24-9
The Replica Subentry	24-10
The Replication Agreement Entry	24-10
The Replication Naming Context Container Entry.....	24-11
The Replication Naming Context Object Entry	24-11
Examples of Replication Configuration Objects in the Directory.....	24-12
Replication Security.....	24-14

Authentication and the Directory Replication Server	24-14
Secure Sockets Layer (SSL) and Oracle Internet Directory Replication	24-15
Change Logs in Directory Replication	24-15
Multimaster Replication	24-16
Oracle Database Advanced Replication	24-16
Architecture for Multimaster Replication	24-17
The Multimaster Replication Process on the Supplier Side	24-17
The Multimaster Replication Process on the Consumer Side	24-18
Conflict Resolution in Multimaster Replication	24-19
Levels at Which Replication Conflicts Occur	24-19
Typical Causes of Conflicts	24-20
Automated Resolution of Conflicts	24-20
Fan-Out and Partial Replication	24-21
Rules for Oracle Database Advanced Replication Filtering	24-22
Rules for Partial Replication Filtering	24-22
Rules for Managing Naming Contexts and Attributes	24-27
Optimization of Partial Replication Naming Context for Better Performance	24-28

25 Oracle Internet Directory Replication Administration

Installing and Configuring Multimaster Replication	25-1
Rules for Configuring Directory Replication Based on Oracle Database Advanced Replication	25-1
Installing and Configuring a Multimaster Replication Group	25-3
Preliminary Information for Installing and Configuring a Multimaster Replication Group	25-3
Task 1: Install Oracle Internet Directory as a Master on the Master Definition Site (MDS)	25-6
Task 2: Install the Oracle Internet Directory as a Replica, on the Remote Master Sites (RMS)	25-6
Task 3: Set Up Oracle Database Advanced Replication for a Directory Replication Group	25-7
Task 4 (Optional): Load Data into the Directory	25-10
Task 5: Ensure that Oracle Directory Server Instances are Started on All the Nodes .	25-11
Task 6: Start the Replication Servers on All Nodes in the DRG	25-11
Task 7: Test Directory Replication	25-12
Adding a Node for Multimaster Replication (Oracle Database Advanced Replication Types Only)	25-13
Prepare the Oracle Net Services Environment	25-13
Task 1: Stop the Directory Replication Server on All Nodes	25-14
Task 2: Identify a Sponsor Node and Install Oracle Internet Directory as a Replica on the Remote Site	25-14
Task 3: Switch the Sponsor Node to Read-Only Mode	25-14
Task 4: Back up the Sponsor Node by Using Idifwrite	25-14
Task 5: Perform Advanced Replication Add Node Setup	25-15
Task 6: Switch the Sponsor Node to Updatable Mode	25-15
Task 7: Start the Directory Replication Server on All Nodes Except the New Node...	25-16
Task 8: Load Data into the New Node by Using bulkload	25-16
Task 9: Start the Directory Server on the New Node	25-16

Task 10: Start the Directory Replication Server on the New Node.....	25-16
Deleting a Node from a Multimaster Replication Group	25-17
Task 1: Stop the Directory Replication Server on All Nodes.....	25-17
Task 2: Stop All Oracle Internet Directory Processes in the Node to be Deleted	25-17
Task 3: Delete the Node from the Master Definition Site	25-17
Task 4: Start the Directory Replication Server on All Nodes	25-18
Resolving Conflicts Manually in a Multimaster Replication Group	25-18
Monitoring Replication Change Conflicts.....	25-18
Examples of Conflict Resolution Messages.....	25-18
About the Human Intervention Queue Manipulation Tool	25-19
About the Oracle Internet Directory Reconciliation Tool	25-19
Installing and Configuring LDAP-Based Replication	25-20
Rules for Configuring LDAP-Based Replication.....	25-20
Back Up Your LDAP Data by Using Idifwrit and bulkload	25-21
Installing and Configuring an LDAP Replica with Default Settings.....	25-21
Task 1: Identify and Start the Directory Server on the Supplier Node	25-22
Task 2: Installing Oracle Internet Directory As An LDAP Replica.....	25-22
Installing and Configuring an LDAP-Based Replica with Customized Settings	25-22
Configuring an LDAP-Based Replica by Using Automatic Bootstrapping	25-23
Configuring an LDAP-Based Replica by Using the Idifwrit Tool	25-27
Deleting an LDAP-Based Replica	25-31
Task 1: Stop the Directory Replication Server on the Node to be Deleted	25-31
Task 2: Delete the Replica from the Replication Group	25-31
Task 3: Stop the Directory Server on the Node to be Deleted	25-31
Determining What Is to Be Replicated in LDAP-Based Partial Replication.....	25-32
Viewing and Modifying Replica Naming Context Objects by Using Oracle Directory Manager.....	25-32
Adding Replica Naming Context Objects by Using Oracle Directory Manager	25-32
Deleting Replica Naming Context Objects by Using Oracle Directory Manager.....	25-32
Modifying Replica Naming Context Object Parameters by Using ldapmodify	25-32
Managing Replication	25-35
Viewing and Modifying Directory Replication Server Configuration Parameters.....	25-35
Viewing Configuration Parameters of the Directory Replication Server by Using Oracle Directory Manager.....	25-35
Modifying Configuration Parameters of the Directory Replication Server by Using Oracle Directory Manager.....	25-36
Modifying Directory Replication Server Configuration Parameters by Using Command-Line Tools.....	25-36
Viewing and Modifying Parameters for Particular Replica Nodes	25-37
Viewing and Modifying Parameters for a Particular Replica Node by Using Oracle Directory Manager.....	25-37
Modifying a Particular Replica Node by Using Command-Line Tools.....	25-38
Modifying Parameters for Replication Agreements	25-39
Modifying Parameters for Replication Agreements Based on Oracle Database Advanced Replication	25-39
Modifying Parameters for Replication Agreements Based on LDAP	25-41
Changing the Replication Administrator's Password on All Nodes.....	25-43
Managing the Change Log.....	25-43

Modifying the Speed of Directory Replication	25-44
Modifying the Speed of Directory Replication When Using Oracle Database Advanced Replication	25-44
Modifying the Speed of Directory Replication When Using LDAP-Based Replication	25-44
Example: Installing and Configuring a Multimaster Replication Group with Fan-Out	25-45

Part VI Directory Plug-ins

26 Oracle Internet Directory Plug-in Framework

About Directory Server Plug-ins.....	26-1
Creating Plug-ins.....	26-3
Registering and Managing Plug-ins.....	26-3
Registering and Managing Plug-ins by Using Oracle Directory Manager.....	26-3
Adding a Plug-in Configuration Entry by Using Oracle Directory Manager.....	26-4
Editing a Plug-in by Using Oracle Directory Manager	26-4
Deleting a Plug-in by Using Oracle Directory Manager	26-4
Registering and Managing Plug-ins by Using Command-Line Tools	26-4
Examples: Adding a Plug-in Configuration Entry by Using Command-Line Tools.....	26-5
Example: Modifying a Plug-in Configuration Entry by Using Command-Line Tools .	26-5
Example: Deleting a Plug-in Configuration Entry by Using Command-Line Tools	26-6

27 Oracle Internet Directory Plug-In for Password Policies

How the Password Policy Plug-in Works	27-1
Example: Installing, Configuring, and Enabling a Customized Password Policy Plug-in.....	27-2
Loading and Registering the PL/SQL Program.....	27-3
Coding the Password Policy Plug-in.....	27-3
Debugging the Password Policy Plug-in.....	27-4
Contents of Sample PL/SQL Package pluginpkg.sql.....	27-4

28 Setting Up the Customized External Authentication Plug-in

Native Authentication Contrasted with External Authentication	28-1
Example: Installing, Configuring, and Enabling the External Authentication Plug-in	28-2
Sample PL/SQL Package oidexaup.sql	28-2
Debugging the External Authentication Plug-in.....	28-3
Contents of PL/SQL Package oidexaup.sql.....	28-4

Part VII Appendixes

A Windows and Fields in Oracle Directory Manager

Connection Management Fields in Oracle Directory Manager	A-2
Access Control Management Fields in Oracle Directory Manager	A-4
Attribute Uniqueness Fields in Oracle Directory Manager.....	A-5
Garbage Collection Management Fields in Oracle Directory Manager	A-6
Oracle Internet Directory Statistics Collection Management Fields in Oracle Directory Manager.....	A-6

Password Policy Fields in Oracle Directory Manager	A-6
Password Verifier Fields in Oracle Directory Manager	A-8
Plug-in Management Fields in Oracle Directory Manager	A-9
Replication Fields in Oracle Directory Manager	A-11
Schema Management Fields in Oracle Directory Manager	A-14
Object Classes Fields in Oracle Directory Manager	A-14
Attributes Fields in Oracle Directory Manager	A-16
Matching Rules Fields in Oracle Directory Manager.....	A-18
Content Rules Management Fields in Oracle Directory Manager.....	A-19
Server Management Fields in Oracle Directory Manager	A-20
Configuration Sets Fields in Oracle Directory Manager	A-20
System Operational Attributes Fields in Oracle Directory Manager	A-22
Super, Guest, and Proxy User Fields in Oracle Directory Manager	A-25
Query Optimization Fields in Oracle Directory Manager	A-26
Entry Search Fields and Buttons in Oracle Directory Manager	A-27
SSL Management Fields in Oracle Directory Manager	A-28
Synchronization Fields in Oracle Directory Manager	A-29

B The LDAP Filter Definition

C The Access Control Directive Format

Schema for orclACI	C-1
Schema for orclEntryLevelACI	C-2

D Globalization Support in the Directory

About Character Sets and the Directory	D-1
About Unicode.....	D-2
About Oracle and UTF-8.....	D-2
Migration from UTF8 to AL32UTF8 when Upgrading Oracle Internet Directory	D-2
The NLS_LANG Environment Variable	D-3
Using Non-AL32UTF8 Databases	D-4
Using Globalization Support with LDIF Files	D-4
An LDIF file Containing Only ASCII Strings.....	D-4
An LDIF file Containing UTF-8 Encoded Strings.....	D-4
CASE 1: Native Strings (Non-UTF-8).....	D-5
CASE 2: UTF-8 Strings	D-5
CASE 3: BASE64 Encoded UTF-8 Strings.....	D-5
CASE 4: BASE64 Encoded Native Strings	D-5
Using Globalization Support with Command-Line Tools	D-5
Specifying the -E Argument When Using Each Tool.....	D-5
Examples: Using the -E Argument with Command-Line Tools	D-6
Setting NLS_LANG in the Client Environment	D-7
Using Globalization Support with Bulk Tools	D-7
Using Globalization Support with bulkload.....	D-8
Using Globalization Support with ldifwrite	D-8
Using Globalization Support with bulkdelete	D-9

Using Globalization Support with bulkmodify	D-9
---	-----

E Setting up Access Controls for Creation and Search Bases for Users and Groups

Setting up Access Controls for the User Search Base and the User Creation Base	E-1
Setting up Access Controls for the Group Search Base and the Group Creation Base	E-3

F The Multimaster Replication Process

How the Multimaster Replication Process Adds a New Entry to a Consumer	F-1
How the Multimaster Replication Process Deletes an Entry	F-2
How the Multimaster Replication Process Modifies an Entry	F-2
How the Multimaster Replication Process Modifies a Relative Distinguished Name	F-3
How the Multimaster Replication Process Modifies a Distinguished Name	F-4

G Searching the Directory for User Certificates

Certificate Mapping	G-1
Search Types	G-2

H LDAP Replica States

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

Definitions	I-1
Prerequisites	I-1
Sponsor Directory Site Environment	I-2
New Directory Site Environment	I-2
Tasks To Be Performed on the New Node	I-2
Tasks To Be Performed on the Sponsor Node	I-3
Tasks To Be Performed on the New Node	I-9
Verification Process	I-13

J Troubleshooting Oracle Internet Directory

Problems and Solutions	J-1
Installation Errors	J-1
TCP/IP Problems	J-1
Do Not Use TCP-Based Monitoring of Oracle Internet Directory Server Availability on Microsoft Windows 2003 Server	J-2
Do Not Install DaimondCS Port Explorer	J-2
Directory Server Error Messages and Causes	J-2
Oracle Database Server Error Due to Schema Modifications	J-2
Constraint Violation Error Due to Editing a User or Group or Creating a Realm	J-2
Standard Error Messages Returned from Oracle Directory Server	J-2
Additional Directory Server Error Messages	J-4
Troubleshooting Password Policies	J-7
Password Policy Error Messages	J-7
Possible Password Policy Problems	J-7

Troubleshooting Directory Performance.....	J-9
Poor LDAP Search Performance.....	J-9
Poor LDAP Add or Modify Performance.....	J-9
Troubleshooting Starting, Stopping, and Restarting of the Directory Server	J-10
About the Tools for Starting, Stopping, and Restarting the Directory Server Instance .	J-10
Problems Starting, Stopping, and Restarting the Directory Server	J-11
Troubleshooting Oracle Internet Directory Replication.....	J-15
Replication Server Does Not Start	J-15
Repository Creation Assistant Error	J-17
Errors in Replication Bootstrap.....	J-17
Changes Are Not Replicated.....	J-19
Replication Stops Working	J-20
Troubleshooting SSL Setup.....	J-20
Troubleshooting Change Log Garbage Collection.....	J-21
Change Logs Are Not Purged.....	J-21
Troubleshooting Dynamic Password Verifiers.....	J-22
Troubleshooting Oracle Internet Directory Password Wallets	J-22
Oracle Internet Directory Server Does Not Start.....	J-22
Password Not Synchronized.....	J-23
Troubleshooting bulkload.....	J-23
Need More Help?	J-24

Glossary

Index

List of Figures

1-1	Oracle Internet Directory Overview	1-6
2-1	A Typical Oracle Internet Directory Node.....	2-2
2-2	Oracle Directory Server Instance Architecture.....	2-4
2-3	A Directory Information Tree	2-7
2-4	Attributes of the Entry for Anne Smith	2-9
2-5	Correct and Incorrect Naming Contexts.....	2-14
2-6	A Replicated Directory.....	2-18
2-7	A Partitioned Directory.....	2-19
2-8	Using Knowledge References to Point to Naming Contexts.....	2-20
2-9	Oracle Identity Management Infrastructure and Other Components	2-24
2-10	Placement of Resource Access and Resource Type Information in the DIT	2-27
5-1	Directory Entry Hierarchy Showing Multiple Configuration Set Entries	5-2
5-2	Alias Entries Example	5-12
5-3	Resulting Tree when Creating the My_file.ldif	5-13
5-4	A Client Locating a Directory Server by Using DNS.....	5-17
7-1	Example of a Directory Information Tree	7-3
10-1	Sample Audit Log in DSE.....	10-9
10-2	Architecture of Oracle Internet Directory Server Manageability.....	10-14
12-1	Indirect Authentication	12-5
14-1	Structural Access Item: Added Object Filter Tab Page.....	14-25
14-2	Structural Access Item: By Whom Tab Page.....	14-26
14-3	Example: Structural Access Item: Access Rights Tab Page.....	14-26
14-4	Content Access Item: By Whom Tab Page	14-27
14-5	Content Access Item: Attribute Tab Page.....	14-28
14-6	Content Access Item: Access Rights Tab Page.....	14-28
14-7	Content Access Item: By Whom Tab Page	14-29
14-8	Content Access Item: Attribute Tab Page.....	14-30
14-9	Content Access Item: Access Rights Tab Page.....	14-30
14-10	Content Access Item: By Whom Tab Page	14-31
14-11	Content Access Item: Attribute Tab Page.....	14-32
14-12	Content Access Item: Access Rights Tab Page.....	14-32
14-13	Content Access Item: By Whom Tab Page	14-33
14-14	Content Access Item: Attribute Tab Page.....	14-34
14-15	Content Access Item: Access Rights Tab Page.....	14-34
15-1	Location of Password Policy Entries.....	15-3
16-1	Location of the Password Verifier Profile Entry	16-5
16-2	Authentication Model	16-7
16-3	How Password Verification Works.....	16-9
17-1	Delegation Flow in an Oracle Application Server Environment	17-2
19-1	Planning the Directory Information Tree	19-2
19-2	Example of an Identity Management Realm	19-7
19-3	Enterprise Use Case: Single Identity Management Realm	19-8
19-4	Enterprise Use Case: Multiple Identity Management Realms	19-8
19-5	Hosted Deployment Use Case	19-9
19-6	Default Identity Management Realm.....	19-10
20-1	Usage Analysis of Current E-mail System	20-4
22-1	Example: Garbage Collection of Change Log Entries.....	22-4
22-2	Garbage Collection Entries in the DIT	22-5
23-1	Using an LDIF File and bulkload	23-3
23-2	Using dipassistant Directly.....	23-4
23-3	Using an LDIF File and dipassistant	23-5
23-4	Using dipassistant, bulkload, and LDIF Files.....	23-5
23-5	Using the Oracle Directory Integration and Provisioning Server.....	23-6
23-6	Structure of the Intermediate User File.....	23-9

24-1	Example of Partial Replication.....	24-3
24-2	Example of Single-Master Replication.....	24-4
24-3	Example of Multimaster Replication	24-5
24-4	Example of Fan-Out Replication.....	24-6
24-5	Example of Multimaster Replication with Fan-Out.....	24-7
24-6	Example of a Naming Context Container and Its Objects	24-8
24-7	Example: Multimaster Replication and Fan-Out Replication	24-12
24-8	Example: Replication Configuration Entries for Node C.....	24-13
24-9	Example: Replication Configuration Entries for Node D	24-14
24-10	The Multimaster Replication Process on the Supplier Side.....	24-17
24-11	The Multimaster Replication Process on the Consumer Side	24-18
24-12	The Fan-Out Replication Process.....	24-21
24-13	A Sample Naming Context.....	24-23
24-14	Naming Context Object #1.....	24-24
24-15	Naming Context Object #2.....	24-24
24-16	Result of Combining Naming Context Objects #1 and #2.....	24-25
24-17	Naming Context Object #3.....	24-26
24-18	Naming Context Object #4.....	24-26
24-19	Result of Combining Naming Context Objects #3 and #4.....	24-27
24-20	Naming Context Object #5.....	24-29
24-21	Naming Context Object #6.....	24-29
24-22	Naming Context Object #7.....	24-30
25-1	Example of Fan-Out Replication.....	25-46
26-1	Oracle Internet Directory Plug-in Framework	26-2

List of Tables

1-1	Comparison of Online Directories and Relational Databases	1-2
2-1	Components of an Oracle Internet Directory Node.....	2-3
2-2	Attributes Created with Each New Entry	2-10
2-3	Common LDAP Attributes.....	2-10
3-1	Tasks to Reset the Default Security Configuration	3-2
3-2	Log File Locations	3-4
4-1	Operating System-Specific Instructions for Starting Oracle Directory Manager	4-2
4-2	Oracle Directory Manager Menu Bar.....	4-4
4-3	Oracle Directory Manager Toolbar	4-5
4-4	Task Areas in Oracle Directory Manager	4-7
4-5	Process Control Items in the ODS_PROCESS Table.....	4-13
4-6	Tools for Starting, Stopping, and Monitoring Oracle Internet Directory Servers	4-16
4-7	Tools for Managing Entries	4-16
4-8	Command-Line Tools for Performing Bulk Operations	4-17
4-9	Command-Line Tools for Managing Replication.....	4-18
4-10	Routine Administration Tasks	4-19
5-1	Definitions of Super User, Guest User, and Proxy User	5-9
5-2	Names, Passwords, and Attributes for Super, Guest, and Proxy Users	5-10
5-3	Orclanonymousofbindflag Value and Directory Server Behavior.....	5-10
5-4	Entry Alias Dereferencing Messages	5-15
5-5	Arguments in a Service Location Record (SRV).....	5-19
6-1	Command-Line Tools for Managing Entries	6-7
7-1	Attribute Uniqueness Constraint Entry.....	7-2
8-1	Content Rule Parameters	8-19
9-1	orclDynamicGroup Attributes for "Connect By" Assertions.....	9-3
9-2	Static and Dynamic Group Considerations	9-5
10-1	Fields in Trace Messages.....	10-4
10-2	Debug Logging Levels	10-5
10-3	Debug Dimension Values for LDAP Operations	10-5
10-4	Attributes of the orclAuditoc Object Class.....	10-8
10-5	Auditable Events.....	10-9
10-6	Audit Mask Levels.....	10-10
10-7	Example: Setting the Audit Level.....	10-11
10-8	Components of Oracle Internet Directory Server Manageability	10-15
10-9	Critical Event Levels.....	10-16
10-10	Fields in the Start a New LDAP Server Instance Window of the Application Server Control Console.....	10-17
10-11	Fields in the Restart an LDAP Server Instance Window of the Application Server Control Console.....	10-18
13-1	SSL Cipher Suites Supported in Oracle Internet Directory	13-1
14-1	Sample Security Groups.....	14-5
14-2	Types of Access	14-9
14-3	LDAP Operations and Access Needed to Perform Each One	14-10
14-4	Attribute States During ACL Evaluation	14-10
14-5	DNs Used in Example	14-38
15-1	Tasks and Tools for Managing Password Polices	15-4
16-1	Attributes for Storing Password Verifiers in User Entries.....	16-5
17-1	Default Privileges Granted to Everyone and to Each User.....	17-3
17-2	Privileges for Administering the Oracle Technology Stack.....	17-4
17-3	Characteristics of the Subscriber DAS Create User Group.....	17-5
17-4	Characteristics of the Subscriber DAS Edit User Group.....	17-6
17-5	Characteristics of the DAS Delete User Group.....	17-6
17-6	Characteristics of the User Privilege Assignment Group	17-6

17-7	Characteristics of the Group Creation Group.....	17-7
17-8	Characteristics of the Group Edit Group.....	17-7
17-9	Characteristics of the Group Delete Group.....	17-8
17-10	Characteristics of the Group Privilege Assignment Group.....	17-8
17-11	Characteristics of the Oracle Application Server Administrators Group.....	17-9
17-12	Characteristics of the User Management Application Administrators Group.....	17-10
17-13	Characteristics of the Trusted Application Administrators Group.....	17-10
17-14	Characteristics of the User Security Administrators Group.....	17-12
17-15	Characteristics of the Authentication Services Group.....	17-12
17-16	Characteristics of the Verifier Services Group.....	17-12
17-17	Characteristics of the User Proxy Privilege Group.....	17-13
17-18	Characteristics of the Oracle Context Administrators Group.....	17-13
17-19	Characteristics of the Common User Attributes Group.....	17-14
17-20	Characteristics of the Common Group Attributes Group.....	17-14
17-21	Characteristics of the Service Registry Viewers Group.....	17-14
17-22	Characteristics of the Common Group Attributes Group.....	17-14
18-1	CPU Power for Various Deployment Scenarios.....	18-7
18-2	Approximate Disk Space Requirements for Variously Sized DITs.....	18-7
18-3	Estimates of the Memory Requirements for Various DIT Sizes.....	18-7
19-1	Oracle Identity Management Objects.....	19-9
19-2	Customizing the Default Identity Management Realm.....	19-12
20-1	Capacity Planning Terminology.....	20-2
20-2	Assumptions about Entry Types and Their Sizes.....	20-2
20-3	Overall Count of Entries.....	20-3
20-4	Directory Lookups in a Single Day.....	20-3
20-5	Working Hour Loads.....	20-4
20-6	Disk Space Requirements.....	20-5
20-7	Tablespaces Used to Store Oracle Internet Directory Data.....	20-6
20-8	Variables Used for Size Calculation.....	20-6
20-9	Size of Individual Tablespaces.....	20-7
20-10	Values for Variables Used for Sizing Calculations.....	20-8
20-11	Tablespace Sizes.....	20-9
20-12	Minimum Memory Requirements for Different Directory Configurations.....	20-10
20-13	Maximum Possible Throughput for Two Types of Operations.....	20-10
20-14	Rough Estimates of CPU Requirements.....	20-12
21-1	Suggested Values for ORCLSERVERPROCS and ORCLMAXCC Parameters.....	21-3
21-2	Recommended RDBMS Values for Various Client Loads.....	21-7
23-1	Features of bulkload and dipassistant.....	23-3
23-2	Mandatory Attributes in a User Entry.....	23-10
24-1	Types of Directory Replication Groups.....	24-2
24-2	Comparison of Full and Partial Replication.....	24-3
24-3	Types of Data Transfer Between Nodes in a Directory Replication Group.....	24-4
24-4	Multimaster, Single-Master, and Fan-Out Replication Compared.....	24-6
24-5	Types of Replication Conflict.....	24-20
25-1	Data Migration Using Idifwrite/bulkload versus Automatic Bootstrapping.....	25-23
25-2	Nodes in Example of Partial Replication Deployment.....	25-45
26-1	Types of Operation-Based Plug-ins.....	26-2
A-1	Fields in the Credentials Tab Page.....	A-2
A-2	Fields in the SSL Tab Page.....	A-3
A-3	Fields in the Access Control Management Pane.....	A-4
A-4	Fields in Authentication Choice List.....	A-4
A-5	Fields in Encryption Choice List.....	A-4
A-6	Entities to Whom You Are Granting Access in the By Whom Tab Page.....	A-4
A-7	Access Rights for Attributes.....	A-5
A-8	Fields in the New Constraint Dialog Box.....	A-5

A-9	Fields in the Garbage Collector Window	A-6
A-10	Fields in the Password Policies General Tab Page.....	A-6
A-11	Fields in the Password Policies Account Lockout Tab Page	A-8
A-12	Fields in the Password Policies IP Lockout Tab Page	A-8
A-13	Fields in the Password Policies Password Syntax Tab Page	A-8
A-14	Fields in the Password Verifier Profile Dialog Box.....	A-8
A-15	Fields in the New Plug-in Dialog Box.....	A-9
A-16	Fields in the Replication Server Configuration Set: General Tab Page.....	A-11
A-17	Fields in the ASR Agreement Tab Page.....	A-11
A-18	Fields in the Replica Node: General Tab Page.....	A-12
A-19	Columns in the Replica Agreements Tab Page	A-12
A-20	Fields in the Replica Naming Context Tab Page.....	A-13
A-21	Fields in the Change Log Window	A-13
A-22	Object Class Properties Listed in Searches in Oracle Directory Manager	A-14
A-23	Search Filters for Object Classes	A-15
A-24	Buttons Used in Searches for Object Classes in Oracle Directory Manager.....	A-15
A-25	Fields in the New Object Class Dialog Box.....	A-16
A-26	Columns in the Attributes Tab Page in Oracle Directory Manager	A-16
A-27	Search Filters for Attributes.....	A-17
A-28	Buttons in Searches for Attributes in Oracle Directory Manager	A-17
A-29	Fields in the General Tab Page of the New Attribute Type Dialog.....	A-17
A-30	Fields in the Advanced Tab Page of the New Attribute Type Dialog.....	A-18
A-31	Fields in the Matching Rules Tab Page.....	A-18
A-32	Fields in the New Content Rule Dialog Box	A-19
A-33	Fields in the Content Rule Dialog Box.....	A-19
A-34	Fields in the Configuration Sets Dialog Box—General Tab Page	A-21
A-35	Fields in the Configuration Sets—SSL Settings Tab Page.....	A-21
A-36	System Operation Attributes Displayed in Oracle Directory Manager.....	A-22
A-37	Fields in the System Passwords Tab Page.....	A-25
A-38	Fields in the Query Optimization Tab Page.....	A-26
A-39	Search Filters for Entries	A-27
A-40	Buttons Used in Searches for Entries	A-27
A-41	Fields in the SSL Settings Tab Page.....	A-28
A-42	Fields on the General Tab Page for Synchronization in Oracle Directory Manager.....	A-29
A-43	Fields on the Execution Tab for Synchronization in Oracle Directory Manager.....	A-30
A-44	Fields on the Mapping Tab Page for Synchronization in Oracle Directory Manager ..	A-31
A-45	Fields on the Status Tab Page for Synchronization in Oracle Directory Manager.....	A-31
D-1	Unicode Implementations	D-2
D-2	Components of the NLS_LANG Parameter.....	D-3
D-3	Examples: Using the -E Argument with Command-Line Tools	D-6
H-1	LDAP Replica States.....	H-1
J-1	Standard Error Messages.....	J-3
J-2	Additional Error Messages.....	J-4
J-3	Password Policy Violation Error Messages.....	J-7
J-4	Error Messages for Dynamic Password Verifiers	J-22

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](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

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 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 \(LDAP\)](#).

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. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

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

Accessibility of Code Examples in Documentation

Screen readers 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, some screen readers 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 does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, see:

- Online help available through Oracle Directory Manager, the Oracle Internet Directory Single Sign-On Console, and Oracle Enterprise Manager 10g
- The Oracle Application Server and Oracle Database documentation sets, especially:
 - *Oracle Identity Management Concepts and Deployment Planning Guide*
 - *Oracle Identity Management Integration Guide*
 - *Oracle Identity Management Guide to Delegated Administration*
 - *Oracle Identity Management Application Developer's Guide*
 - *Oracle Application Server Single Sign-On Administrator's Guide*
 - *Oracle Application Server Certificate Authority Administrator's Guide*
 - *Oracle Identity Management User Reference*
 - *Oracle Application Server High Availability Guide*
 - *Oracle Application Server Administrator's Guide*
 - *Oracle Database Administrator's Guide*
 - *Oracle Database Net Services Administrator's Guide*
 - *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide*
 - *Oracle Database Advanced Replication*
 - *Oracle Advanced Security Administrator's Guide*

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 LDATEXT 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

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

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 10g Release 2 \(10.1.2\)](#)
- [New Features Introduced with Oracle Internet Directory 10g \(9.0.4\)](#)
- [About Oracle Internet Directory Release 9.2](#)
- [New Features Introduced with Oracle Internet Directory Release 9.0.2](#)
- [New Features Introduced with Oracle Internet Directory Release 3.0.1](#)
- [New Features Introduced with Oracle Internet Directory Release 2.1.1](#)

New Features Introduced with Oracle Internet Directory 10g Release 2 (10.1.2)

Note:

The following chapters have been moved to *Oracle Application Server High Availability Guide*:

- "High Availability And Failover Considerations"
- "Oracle Application Server Cluster (Identity Management) Configurations"
- "Oracle Application Server Cold Failover Cluster (Identity Management)"
- "The Directory in an Oracle Real Application Clusters Environment"

The following appendixes have been rewritten as chapters in *Oracle Identity Management User Reference*:

- "Syntax for LDIF and Command-Line Tools"
 - "Oracle Internet Directory Schema Elements"
-
-

- **Improved integration with other components**—New features provide better integration with components such as Oracle Collaboration Suite. These features include service-to-service authentication, the service registry, and verifier generation using dynamic parameters.

See Also:

- ["The Service Registry and Service to Service Authentication"](#) on page 2-21
- ["Verifier Generation Using Dynamic Parameters"](#) on page 16-10

- **Support for Certificate Matching Rule**—External authentication using certificates can now take either of two forms: an exact match, in which the subject DN of the client certificate is used to authenticate the user, or a certificate hash, in which the client certificate is hashed and is then compared with a certificate hash stored in the directory.

See Also: ["Direct Authentication"](#) on page 12-3

- **Ease of deployment for Replication**—Replication is now much easier to install, configure, and manage.

See Also:

[Chapter 25, "Oracle Internet Directory Replication Administration"](#)
Oracle Application Server Installation Guide

- **Ease of deployment for Clusters**—Cluster configurations are now much easier to install, configure, and manage.

See Also:

"Oracle Application Server Cluster (Identity Management) Configurations" in
Oracle Application Server High Availability Guide
Oracle Application Server Installation Guide.

- **Enforcing access control for Oracle Internet Directory super user**—The super user is now subject to access control policies like any other user. New ACL keywords allow you to restrict super user access through privileged groups.

See Also: [Chapter 14, "Directory Access Control"](#)

- **Oracle Internet Directory Server Diagnostic Tool**—The OID Diagnostic Tool collects diagnostic information that helps triage issues reported on Oracle Internet Directory.

See Also: The "oiddiag" command-line tool reference in *Oracle Identity Management User Reference*

New Features Introduced with Oracle Internet Directory 10g (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 Oracle Directory Integration and Provisioning and plug-ins.

See Also: The chapter on integration with Microsoft Windows in the *Oracle Identity Management Integration Guide*

- **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 28, "Setting Up the Customized External Authentication Plug-in"](#)
- The chapter on considerations for integrating with third-party connected directories in *Oracle Identity Management Integration Guide*

- **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 21-9

- **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 12-3

- **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 23-8
- The "ldifmigrator" command-line tool reference in *Oracle Identity Management User Reference*

- **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 6-14
- The material on the `ldap_set_option` and the `ldap_get_option` in *Oracle Identity Management 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, "Oracle Internet Directory Replication Concepts"](#)
- [Chapter 25, "Oracle Internet 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
- Super user account lockout requiring reset.
- 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 Oracle 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 "remtool" command-line tool reference in *Oracle Identity Management User Reference*

- **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 "remtool" command-line reference in *Oracle Identity Management User Reference*

- **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: The "bulkload" command-line tool reference in *Oracle Identity Management User Reference*

- **Oracle Application Server Cluster (Identity Management) 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 Oracle Database.

See Also: "Oracle Application Server Cluster (Identity Management) Configurations" in *Oracle Application Server High Availability Guide*

- **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: The chapter on provisioning service concepts in the *Oracle Identity Management Integration Guide*

- **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: The chapter on integration with the Oracle E-Business Suite in the *Oracle Identity Management Integration Guide*

- **Installation of Oracle Internet Directory on Oracle Real Application Clusters**—You can install Oracle Internet Directory on Oracle 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: The chapter about the Oracle Internet Directory Self-Service Console in *Oracle Identity Management Guide to Delegated Administration*

- **Upgrade procedures**

See Also: Oracle Application Server Upgrade and Compatibility Guide 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 Oracle Database. 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 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 21-8

- **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 Oracle Directory Integration and Provisioning (introduced with Oracle Internet Directory v2.1.1.1 in the Oracle8i Release 3 time frame).
 - **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: The chapter on concepts and components in the *Oracle Identity Management Integration Guide*

- **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 12-6 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 7, "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 5-8

- **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 10-12

- **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 26, "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 5-11

- **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: The chapter on the Oracle Internet Directory Self-Service Console in *Oracle Identity Management Guide to Delegated Administration*

- **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: "Oracle Application Server Cold Failover Cluster (Identity Management)" in *Oracle Application Server High Availability Guide*

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

Oracle 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: "The Directory in an Oracle Real Application Clusters Environment" in *Oracle Application Server High Availability Guide*

- **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: "Oracle Application Server Cold Failover Cluster (Identity Management)" in *Oracle Application Server High Availability Guide*

- **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 18-5

- **Oracle Directory Integration and Provisioning**

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: *Oracle Identity Management Integration Guide*

- **Password policy management**

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

See Also:

- ["Protection of User Passwords for Directory Authentication"](#) 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 longer 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 2-11 for a conceptual discussion
- ["Managing Entries with Attribute Options by Using Oracle Directory Manager"](#) on page 6-6
- ["Managing Entries with Attribute Options by Using Command-Line Tools"](#) on page 6-9

- **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 22-5 for a conceptual discussion
- ["Viewing and Modifying Directory Replication Server Configuration Parameters"](#) on page 25-35

- **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 2-9 for a conceptual discussion
- The "ldapsearch" command-line tool reference in *Oracle Identity Management User Reference* 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 Data Repositories"](#)

- **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 8-3 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: The "oidstats.sql" command-line tool reference in *Oracle Identity Management User Reference*

- **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 12-6 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 25-19
- ["About the Oracle Internet Directory Reconciliation Tool"](#) on page 25-19

- **Replication node deletion**

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

Tip: ["Deleting a Node from a Multimaster Replication Group"](#) on page 25-17

- **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 synchronize multiple directories with Oracle Internet Directory.

Note: This feature was replaced in Release 3.0.1 by Oracle Directory Integration and Provisioning. See the chapter on concepts and components in the *Oracle Identity Management Integration Guide*

- **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, "Post-Installation Tasks and Information"](#)
- [Chapter 4, "Directory Administration Tools"](#)

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?](#)
- [What Is the Lightweight Directory Access Protocol \(LDAP\)?](#)
- [Oracle Identity Management](#)
- [What Is Oracle Internet Directory?](#)
- [How Oracle Components Use Oracle Internet Directory](#)

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 as central integration points for multiple applications and services. Online directories are also becoming critical to both e-businesses and hosted environments.

This section contains these topics:

- [The Expanding Role of Online Directories](#)
- [The Problem: Too Many Special-Purpose Directories](#)

The Expanding Role of Online Directories

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

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

- An employee searching for corporate white page 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
<p>Primarily read-focused. Typical use involves a relatively small number of data updates, and a potentially large number of data retrievals.</p>	<p>Primarily write-focused. Typical use involves continuous recording of transactions, with retrievals done relatively infrequently.</p>
<p>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.</p>	<p>Designed to handle large and diverse transactions using many operations on large units of data.</p>
<p>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.</p>	<p>Typically designed to be location-specific. While a relational database can be distributed, it usually resides on a particular database server.</p>
<p>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 <code>person</code> 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.</p>	<p>Designed to store information as rows in relational tables.</p>

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 Version 3](#)

LDAP and Simplified Directory Management

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

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

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

LDAP Version 3

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

- **Globalization Support:** LDAP Version 3 allows servers and clients to support characters used in every language in the world.
- **Knowledge references (also called referrals):** LDAP Version 3 implements a referral mechanism that allows servers to return references to other servers as a result of a directory query. This makes it possible to distribute directories globally by partitioning a [directory information tree \(DIT\)](#) across multiple LDAP servers.

- Security: LDAP Version 3 adds a standard mechanism for supporting **Simple Authentication and Security Layer (SASL)**, 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. These are extra pieces of information carried along with existing operations, altering the behavior of the operation. When a client application passes a control along with the standard LDAP command, the behavior of the commanded operation is altered accordingly. For example, when a client wants to modify meta-information hidden in the directory, it can send the manageDSAIT control along with the LDAP command.
- 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 at: <http://www.ietf.org>
- "Related Documents" on page xxx 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
- "About LDAP Controls" in *Oracle Identity Management User Reference* for a list and description of controls supported by Oracle Internet Directory

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: 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 Oracle Identity Management

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 an Oracle Database.

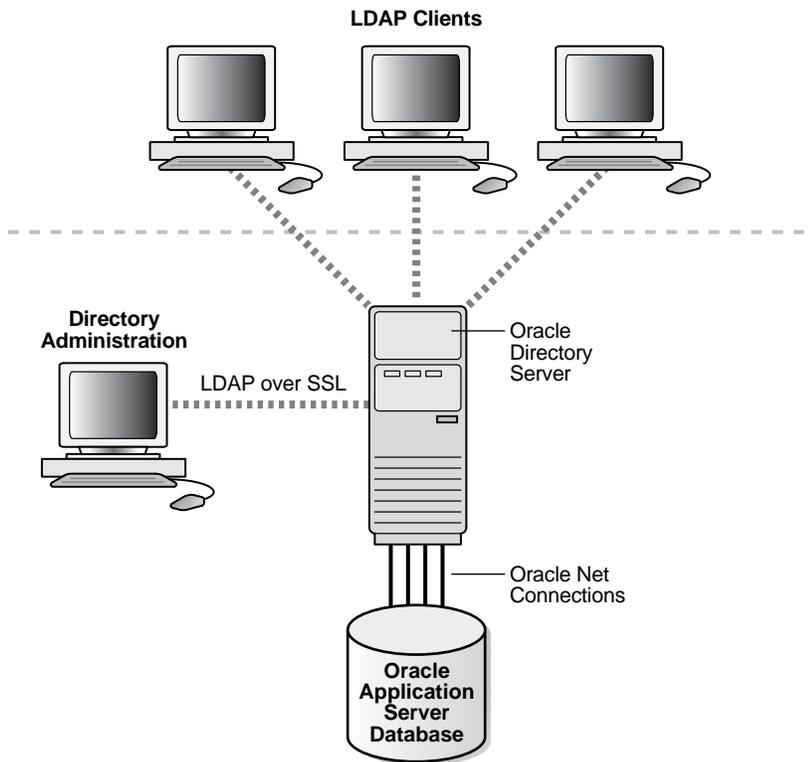
This section contains these topics:

- [Overview of Oracle Internet Directory](#)
- [Components of Oracle Internet Directory](#)
- [Advantages of Oracle Internet Directory](#)

Overview of Oracle Internet Directory

Oracle Internet Directory runs as an application on an Oracle Database. It communicates with the database by using Oracle Net Services, Oracle's operating system-independent database connectivity solution. The database may or may not be on the same host. [Figure 1-1](#) illustrates this relationship.

Figure 1–1 Oracle Internet Directory Overview



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 10g Application Server Control Console. 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 Identity Management 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 an Oracle Database, 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 to directory data on a server is stored in special tables on the Oracle Database. These are replicated throughout the directory environment by [Oracle Database Advanced Replication](#), a robust replication mechanism.

Oracle Internet Directory also takes advantage of all the availability features of the Oracle Database. Because directory information is stored securely in the Oracle Database, it is protected by Oracle's backup capabilities. Additionally, the Oracle Database, running with large data stores 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 Sockets Layer \(SSL\)](#) Version 3 for authenticated access and data privacy.

Integration with the Oracle Environment

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

How Oracle 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 Multiple 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

The **Oracle Database** 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 entries in Oracle Internet Directory, and authenticates those users by using LDAP mechanisms.

Oracle Advanced Security uses Oracle Internet Directory for:

- Central Management of user authentication credentials
Instead of storing a user's database password in each database, Oracle Advanced Security stores it in one place: the directory. It stores the password as an attribute of the user entry.
- Central management of user authorizations
Oracle Advanced Security uses directory entries, called enterprise roles, to determine the privileges for a given enterprise user within a given schema, whether that schema is shared or owned. Enterprise roles are containers for database-specific global roles. For example, a user might be assigned the enterprise role of clerk, which might contain the global role of hrclerk with its attendant privileges on the human resources database and the global role of analyst with 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 schemas 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 the Oracle Database, 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 spares you from setting up

Secure Sockets Layer (SSL) on clients and 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 Oracle Database and Oracle Application Server, user wallets can be stored in the directory as an attribute of the user's entry. This 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 Multiple Directories

Oracle Directory Integration and Provisioning is a collection of interfaces and services for integrating multiple directories by using Oracle Internet Directory and several associated plug-ins and connectors. It 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 saves you from having to integrate each application with each directory.

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:

- [Oracle Internet Directory Architecture](#)
- [Example: How Oracle Internet Directory Works](#)
- [Entries](#)
- [Attributes](#)
- [Object Classes](#)
- [Naming Contexts](#)
- [Security](#)
- [Globalization Support](#)
- [Distributed Directories](#)
- [Knowledge References and Referrals](#)
- [Oracle Delegated Administration Services and the Oracle Internet Directory Self-Service Console](#)
- [The Service Registry and Service to Service Authentication](#)
- [Oracle Directory Integration and Provisioning](#)
- [Oracle Internet Directory and Identity Management](#)
- [Resource Information](#)

See Also: ["Related Documents"](#) on page xxx for suggestions on further reading about LDAP-compliant directories

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 Oracle Database.

Figure 2-1 on page 2-2 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:

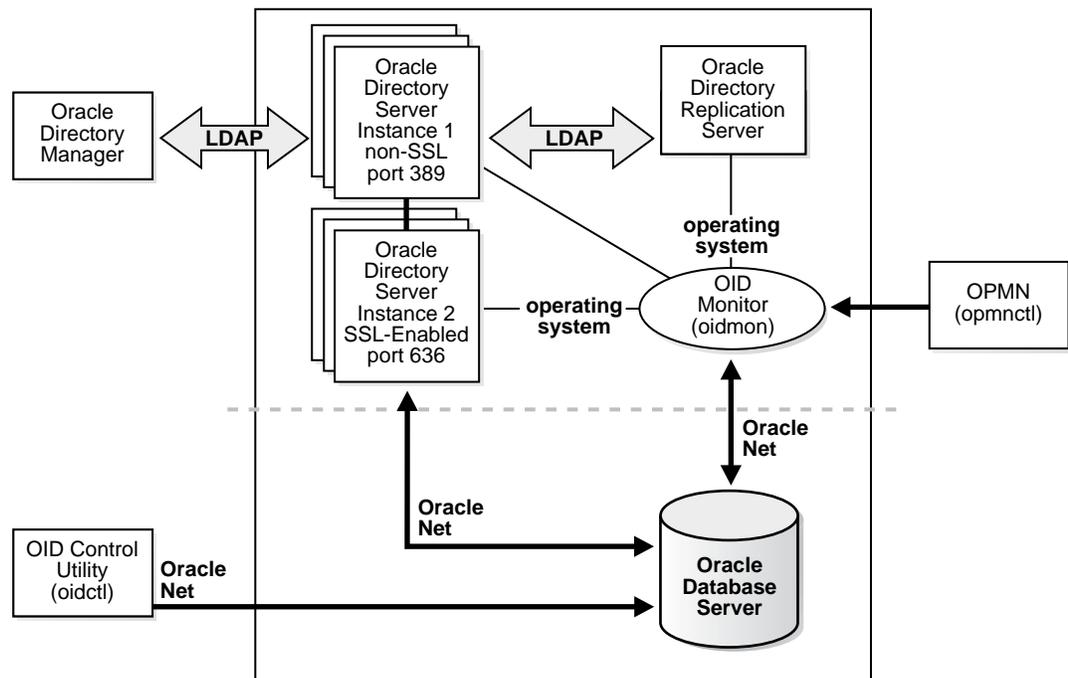
- The **object class**
- The Oracle directory server instance 1 non-SSL port 389
- The Oracle directory server instance 2 SSL-enabled port 636
- The **OID Monitor**

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 2-1 A Typical Oracle Internet Directory Node



As shown in Figure 2-1, an Oracle Internet Directory node includes the following major components:

Table 2–1 Components of an Oracle Internet Directory Node

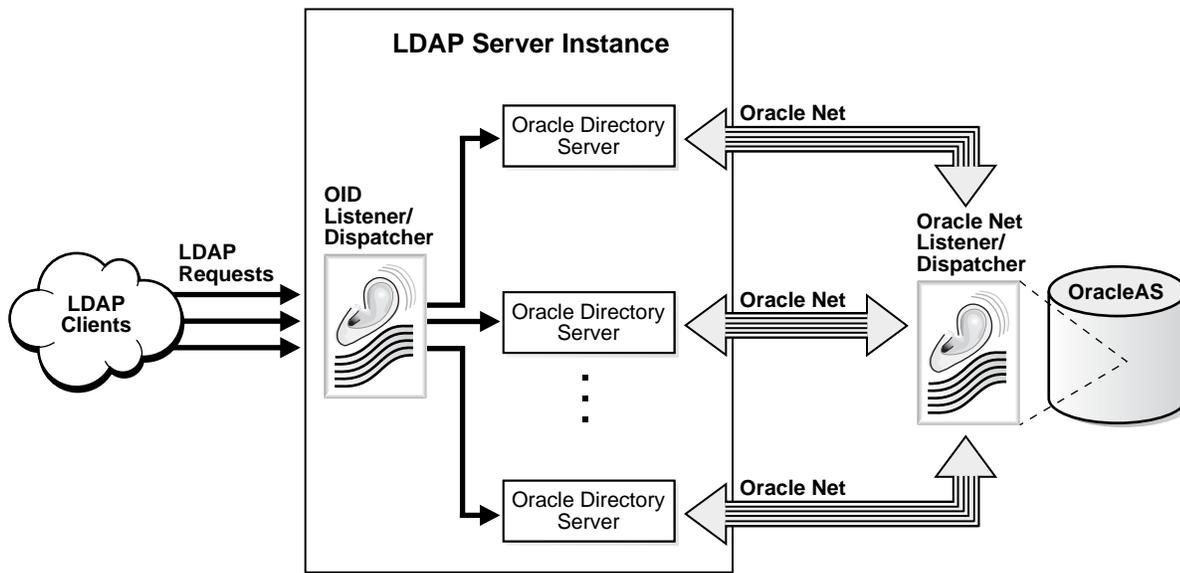
Component	Description
Oracle directory server instance	Also called either an LDAP server instance or a directory server instance, it services directory requests through a single Oracle Internet Directory dispatcher process listening at specific TCP/IP ports. There can be more than one directory server 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.
Oracle Database Server	Stores the directory data. Oracle 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.
Oracle Process Manager and Notification Server (OPMN)	Manages Oracle Internet Directory as an Oracle Application Server component. OPMN uses the directives in the OID component snippet in <code>\$ORACLE_HOME/opmn/conf/opmn.xml</code> and invokes ODMON and OIDCTL as required. It is unaware of the Oracle Internet Directory Server instances.
OID Monitor (OIDMON)	<p>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.</p> <p>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.</p> <p>When it starts a server instance, OID Monitor adds an entry into the directory instance registry and updates data in the process table. It also starts any servers that it finds in the process table. When it shuts down the directory server instance, it updates 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.</p> <p>All OID Monitor activity is logged in the file <code>\$ORACLE_HOME/ldap/log/oidmon.log</code>. This file is on the Oracle Internet Directory server file system.</p> <p>OID Monitor checks the state of the servers through mechanisms provided by the operating system.</p>
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 2–2](#) illustrates.

Figure 2–2 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 Oracle Internet Directory 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 number of database connections spawned by each server is equal to $ORCLMAXCC + (ORCLMAXCC/2) + 1$. The default value of `ORCLMAXCC` in `configset0` is 2. The server processes communicate with the data server by way of Oracle Net Services. An Oracle Net Services Listener/Dispatcher relays the request to the Oracle Database.

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

See Also:

- ["Managing Entries by Using Oracle Directory Manager"](#) on page 6-1
- ["Example: Adding a User Entry by Using Oracle Directory Manager"](#) on page 6-4

- 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 26, "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 5-1
- ["Oracle Internet Directory Configuration Schema Elements"](#) in *Oracle Identity Management User Reference* 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 Oracle Database.
- 6. The Oracle 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.

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.

This section contains these topics:

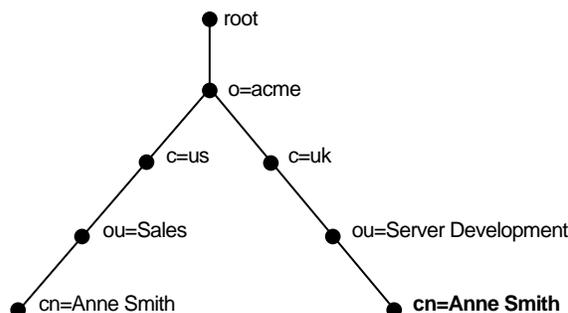
- [Distinguished Names \(DNs\) and Directory Information Trees \(DITs\)](#)
- [Entry Caching](#)

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 2-3](#).

Figure 2-3 A Directory Information Tree



The DIT in [Figure 2-3](#) includes 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 United Kingdom of Great Britain and Northern Ireland (`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 2-3](#), 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 10g Release 2 (10.1.2), you can use entry caching only in the case of a single server, single instance Oracle Internet Directory node.

See Also: [Chapter 6, "Directory Entries Administration"](#)

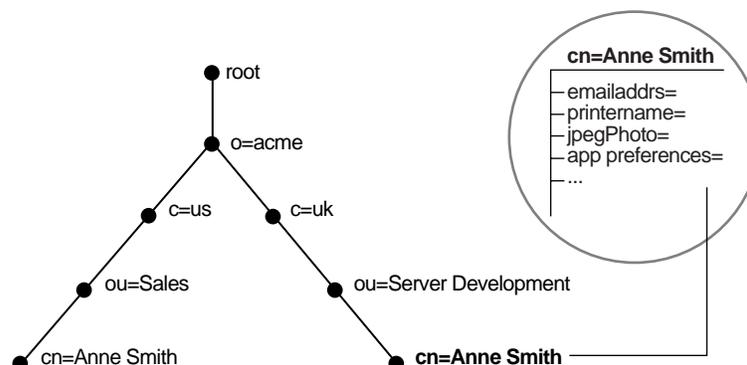
Attributes

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 2-4](#), 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 `emailaddr`, `printername`,

jpegPhoto, and app preferences. Moreover, each bullet in [Figure 2-4](#) is also an entry with attributes, although the attributes for each are not shown.

Figure 2-4 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](#)
- [Common LDAP Attributes](#)
- [Attribute Syntax](#)
- [Attribute Matching Rules](#)
- [Attribute Options](#)

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:

Table 2–2 Attributes Created with Each New Entry

Attribute	Description
creatorsName	Name of the person creating the entry
createTimestamp	Time of entry creation in UTC (Coordinated Universal Time)
modifiersName	Name of person modifying 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 5-7 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. Some of the more common ones defined by RFC 2798 of the Internet Engineering Task Force (IETF) are shown in [Table 2–3](#).

Table 2–3 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: "Oracle Identity Management LDAP Attribute Reference" in *Oracle Identity Management User Reference* 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: "About LDAP Attribute Syntax" in *Oracle Identity Management User Reference*

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: "About LDAP Matching Rules" in *Oracle Identity Management User Reference*

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 6-6
- ["Managing Entries with Attribute Options by Using Command-Line Tools"](#) on page 6-9

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 (ACP)** where this attribute is defined
- `orclEntryLevelACI`—Access control policy pertaining to only a specific entity—for example, a special user

See Also:

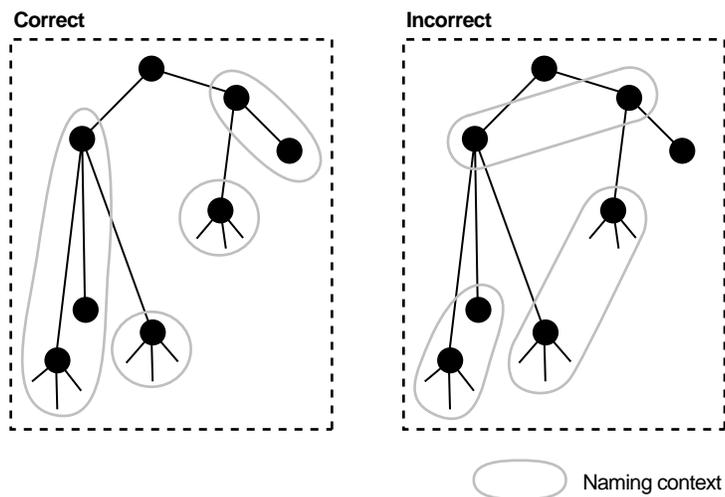
- ["Globalization Support"](#) on page 2-15 for more information on access control policies and ACLs
- ["How to Extend the Number of Attributes Associated with Entries"](#) on page 8-15 for a discussion of how to add additional content to entries

Naming Contexts

A **directory 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 **directory information tree (DIT)**.

Figure 2-5 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 2-5 Correct and Incorrect Naming Contexts



To enable users to discover 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 5-8 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

You can use all these features to enforce a uniform security policy for multiple applications enabled for Oracle Internet Directory, and do so in either an enterprise or hosted environment. You do this by deploying 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
- The chapter on security in the *Oracle Identity Management Integration Guide* for a discussion of the unique aspects of security in an Oracle Directory Integration and Provisioning 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 **UTF-8** (Unicode Transformation Format 8-bit) character set. With Oracle9i, Oracle added a new UTF-8 character set called AL32UTF8. This database character set supports the latest version of Unicode (3.2), including the latest supplementary characters. 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 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 and provisioning 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 2-1 for information on the main Oracle Internet Directory components
- [Appendix D, "Globalization Support in the Directory"](#)
- *Oracle Database Globalization Support Guide* in the Oracle Database Documentation Library for a detailed discussion of Globalization Support

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. Some features of replication are:

- It improves performance by providing more servers to handle queries, and reliability by eliminating risks associated with a single point of failure.
- It 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.

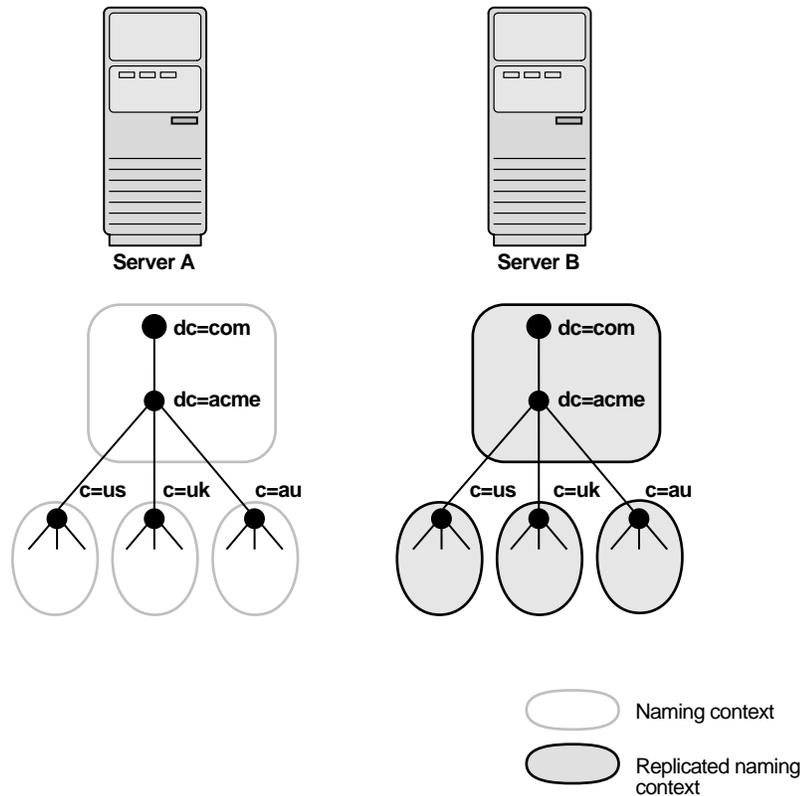
Each copy of a naming context contained within a server is called a replica. Replicas can be read-only, updatable, or both. Servers that hold updatable replicas are called suppliers. Their changes are propagated to other servers called consumers.

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. In a DRG, the protocol for transferring data between nodes can be based on either Oracle Database Advanced Replication or LDAP.

A DRG 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.

Figure 2–6 shows a replicated directory.

Figure 2–6 A Replicated Directory

Note: 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.

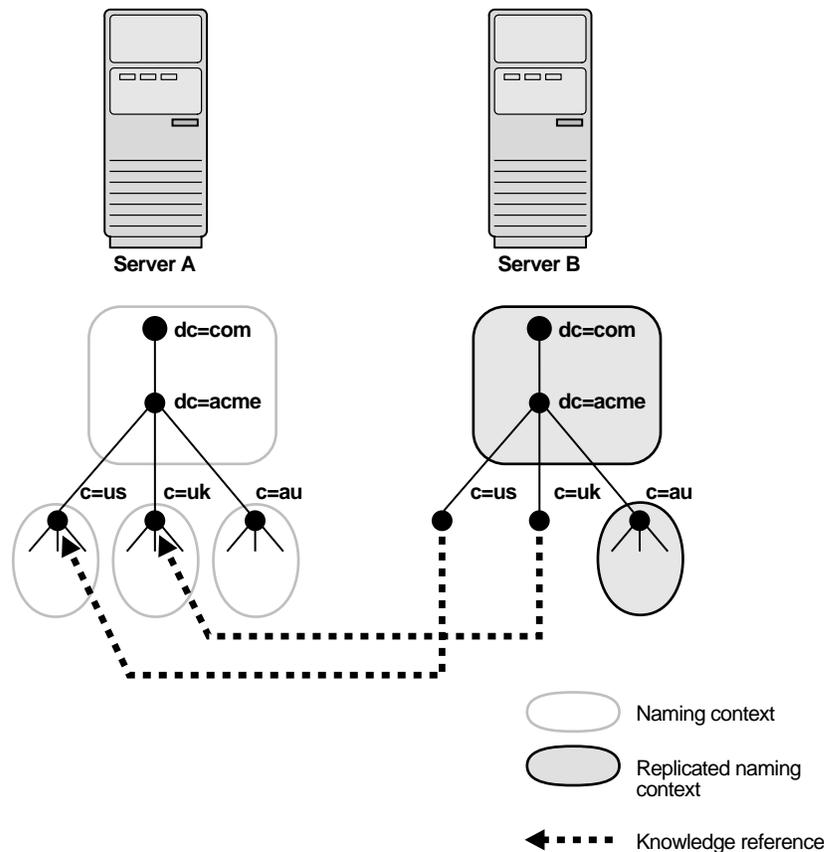
See Also: [Chapter 24, "Oracle Internet Directory Replication Concepts"](#) for a more detailed discussion of replication, including: Oracle Database Advanced Replication architecture, LDAP-based replication, change log purging, conflict resolution, and the replication process

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 2-19, 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

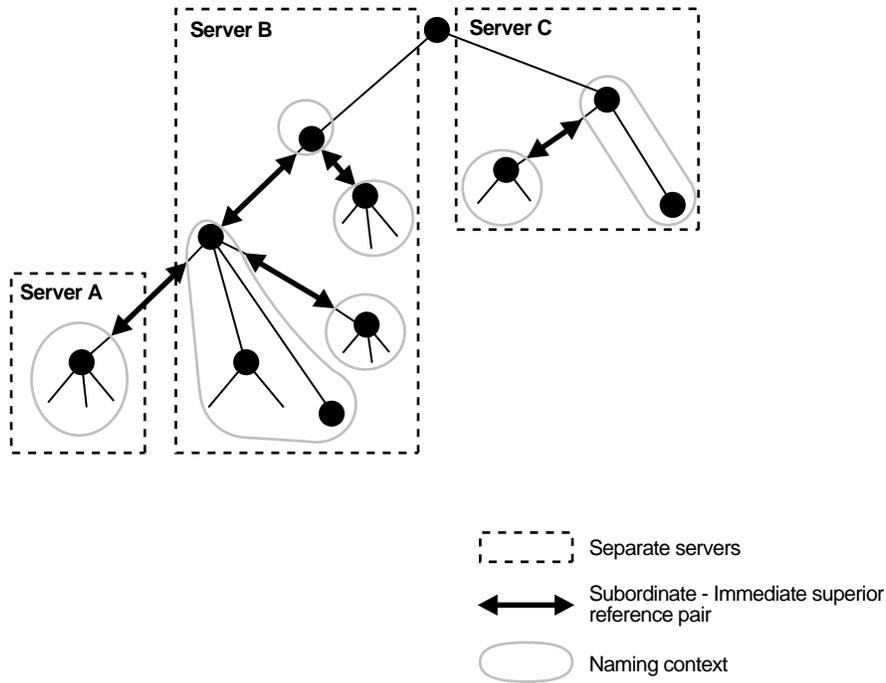
A knowledge reference provides the names and addresses of the various naming contexts held in another partition. For example, in Figure 2-7 on page 2-19, Server B uses knowledge references to point to the c=us and c=uk naming contexts on Server A. When Server B is asked for information residing on Server 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 2-20, 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 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 6-12

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: *Oracle Identity Management Guide to Delegated Administration*

The Service Registry and Service to Service Authentication

The Service Registry and the Service to Service Authentication framework are Oracle Internet Directory features that facilitate integration between Oracle technology components that request services from one another. The Service Registry provides a place to store information, so that the components can discover each other. The Service to Service Authentication framework allows one component to authenticate to another and establishes trust among them.

The Service Registry is a container in Oracle Internet Directory (under `cn=Services`, `Cn=OracleContext`) where components store connectivity information, such as protocol, and other information, such as type of service. During installation, each OCS component registers its information in the Registry. At run-time the components discover information registered by other components. Service Registry objects are stored in the Oracle Internet Directory DIT in a component-specific Services container in the `rootOracleContext`.

Service to Service Authentication is a framework that allows one service to authenticate to another and establish trusts among the services. At install time, each of the client services is provisioned with a username and password in Oracle Internet Directory. In addition, each target service defines an authorization role in Oracle Internet Directory to control which components should it trust. When a component requests services of another component, the requestor must authenticate to the target service like any other client, using its own identity and credentials. The requesting service must also be listed in the Target services Trusted Application group (Default Group: `contrasted Applications, counterpoise, cn=OracleContext`). The requesting service also must send the user's identity so that the target service can authenticate the user as well. The data is sent securely, using either Digest authentication or the target service's native secure authentication.

Oracle Directory Integration and Provisioning

Oracle Directory Integration and Provisioning 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, Oracle Directory Integration and Provisioning 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 Identity Management Integration Guide*

Oracle Internet Directory and Identity Management

Identity management is the process of managing the complete security life cycle for network entities 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. Matching the deployment of Oracle Internet Directory with the security needs of your enterprise requires careful planning.

This section contains these topics:

- [About Identity Management](#)
- [About the Oracle Identity Management Infrastructure](#)
- [Identity Management Realms](#)

About Identity Management

Identity management usually 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.

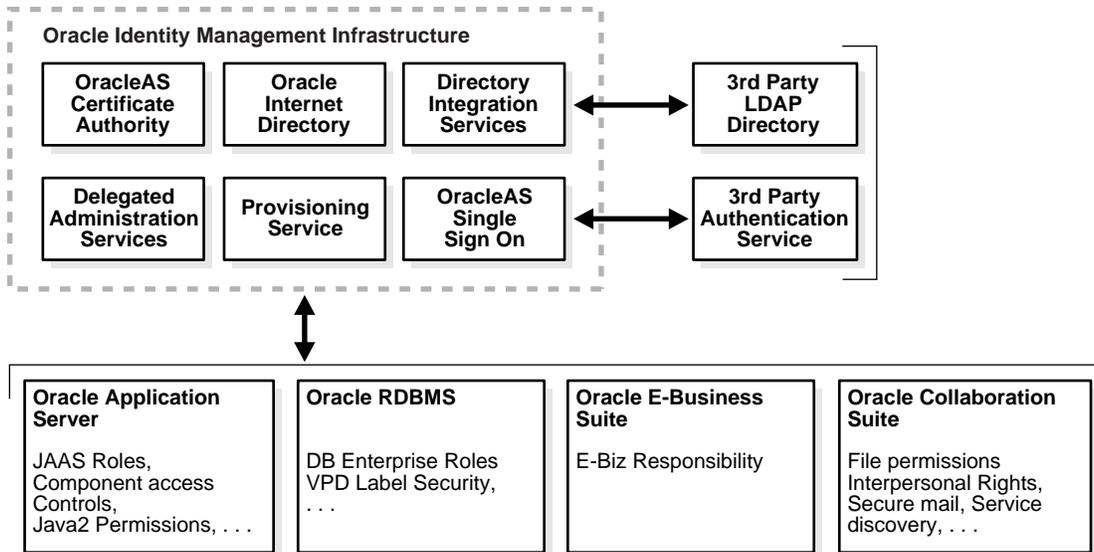
Identity management is important to IT deployments because it can reduce administrative costs while at the same time improving security.

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 2-24 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 Oracle Database.
- Oracle Directory Integration and Provisioning, 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](#)
- [Resource Access Information](#)
- [Location of Resource Information in the DIT](#)

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 Oracle Database 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 Reports 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

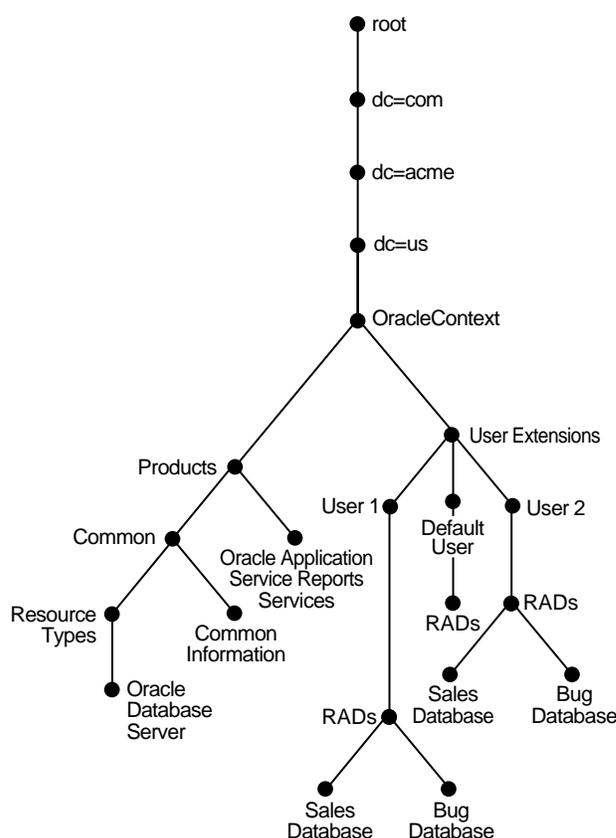
Once it has done this, it compiles the report.

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 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 Reports`,

`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:

- The section about managing your own resource information in *Oracle Identity Management Guide to Delegated Administration* for instructions for an end user to specify resource access information
- The section on creating user entries by using the self-service console in *Oracle Identity Management Guide to Delegated Administration* for instructions for an administrator to specify resource access information when creating user entries
- The section on configuring default resource access information in *Oracle Identity Management Guide to Delegated Administration* for instructions for an administrator to define commonly used resources that all users automatically inherit
- The section on creating a new resource type in *Oracle Identity Management Guide to Delegated Administration* for instructions for an administrator to specify resource types
- "Plug-in Schema Elements" in *Oracle Identity Management User Reference*
- *Oracle Application Server Reports Services Publishing Reports to the Webin Reports* at <http://www.oracle.com/technology/documentation>

Post-Installation Tasks and Information

At the end of a successful installation, the OID Monitor, `oidmon`, and an instance of the Directory Server, `oidldapd`, are running.

Notes:

- You can run multiple instances of the directory server is on the same computer. For example, you can run one instance in SSL mode and another in non-SSL mode.
 - If you restart the computer, you can restart Oracle Application Server Infrastructure by following the steps described in the section "Starting OracleAS Infrastructure" in *Oracle Application Server Administrator's Guide*.
 - The Oracle Internet Directory servers—that is, the directory server, the directory replication server, and the directory integration and provisioning server daemons—can be started only by the operating system user who installed the Oracle Internet Directory software.
-
-

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: Reset the Default Security Configuration](#)
- [Task 2: Reset the Default Password for the Database](#)
- [Task 3: Run the OID Database Statistics Collection Tool](#)
- [Tasks to Perform After Upgrading from Release 9.0.2](#)
- [Determining LDAP Port Assignment on UNIX and Linux](#)
- [Log File Locations](#)

Task 1: 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 recommends that you control access to these objects.
Establish access to entries	When you load directory entries, you are creating a hierarchy of directory entries. You must therefore establish: <ul style="list-style-type: none"> ■ 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.
Modify the password of the super user	The super user has full access to directory information. The default user name of the super user is orcladmin; the default password is the Oracle Application Server Administrator password that was specified during installation. Modify this password immediately after installation.

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
- [Chapter 15, "Password Policies in Oracle Internet Directory"](#) 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 2: Reset the Default Password for the Database

Oracle Internet Directory uses a password when connecting to its designated Oracle database. The default for this password is the same as that specified during installation

for the Oracle Application Server administrator (ias_admin). Change this default password by using the OID Database Password Utility.

See Also: The "oidpasswd" command-line tool reference in *Oracle Identity Management User Reference* for syntax and usage notes

Task 3: 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, \$ORACLE_HOME/ldap/admin/oidstats.sql, after loading. This enables the Oracle Optimizer to choose an optimal plan for executing queries corresponding to 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 8.6 or later. Visit:
<http://www.datafocus.com/>
-

See Also: The "oidstats.sql" command-line reference in *Oracle Identity Management User Reference*

Tasks to Perform After Upgrading from Release 9.0.2

If you have upgraded Oracle Internet Directory from Release 9.0.2 to Release 10.1.2, perform the following tasks.

Set ACL Policy on Groups Container after Upgrade from Release 9.0.2

When upgrading Oracle Internet Directory from Release 9.0.2 to Release 10.1.2, the following ACL policy needs to be set on the groups container in the realm. The ACL policy should allow members of the group `cn=Common Group Attributes,cn=groups,Oracle_Context_DN` browse, search, and read access for private and public groups—that is, for groups where `orclisvisible` is either not set or is set to `TRUE` or `FALSE`. This ACL is described in the *Oracle Internet Directory Administrator's Guide*, in Chapter 17, in the section "Default Privileges for Reading Common Group Attributes".

The "Common Group Attributes" group is used by OracleAS Portal to query private and public groups. The ACI must be added on the groups container. Change the *Realm DN* to the DN of the Realm and the *DN of groups container in the realm* to the appropriate group search base.

```
dn: DN of groups container in the realm
changetype: modify
add: orclaci
orclaci: access to entry filter=(!(orclisvisible=false)) by group="cn=Common Group
Attributes,cn=groups, cn=Oracle Context, Realm DN" (browse)
orclaci: access to attr=(*) filter=(!(orclisvisible=false)) by group="cn=Common
Group Attributes,cn=groups,cn=Oracle Context, Realm DN" (search, read)
orclaci: access to entry filter=(orclisvisible=false) by group="cn=Common Group
```

```
Attributes,cn=groups,cn=Oracle Context, Realm DN" (browse)
orclaci: access to attr=(*) filter=(orclisvisible=false) by group="cn=Common Group
Attributes,cn=groups, cn=Oracle Context, Realm DN" (search, read)
```

Determining LDAP Port Assignment on UNIX and Linux

During installation of Oracle Application Server or third-party products, you might be prompted for an Oracle Internet Directory or LDAP port. To find the specific port number assigned to Oracle Internet Directory at installation, see the file `$ORACLE_HOME/config/ias.properties`. Look for the entries `OIDport` and `OIDsslport`.

The default port for enabling LDAP at Oracle Internet Directory installation time is 389. The Oracle Universal Installer always tries that port as its first choice. However, on many UNIX computers, `/etc/services` includes a line for LDAP reserving port 389. When that line is present, the Installer opts instead for a port number between 3060 to 3129, inclusive.

To confirm the port at which Oracle Internet Directory is running, simply run the `ldapbind` command-line tool, supplying either the host name and port number specified in the `portlist.ini` file or an alternative port specified during the Oracle Internet Directory installation.

Log File Locations

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)	<code>\$ORACLE_HOME/ldap/log/bulkload.log</code>
Catalog Management Tool (catalog.sh)	<code>\$ORACLE_HOME/ldap/log/catalog.log</code>
Directory integration agent	<code>\$ORACLE_HOME/ldap/odi/log/AgentName.err</code> where <i>AgentName</i> is the name of the agent
Directory integration server (odisrv)	<code>\$ORACLE_HOME/ldap/log/odisrvXX.log</code> where <i>XX</i> is Oracle directory integration and provisioning server instance number
Directory replication server (oidrepld)	<code>\$ORACLE_HOME/ldap/log/oidrepld00.log</code>
Directory server (oidldapd)	<code>\$ORACLE_HOME/ldap/log/oidldapdXXspid.log</code> where <i>pid</i> is the server process identifier <code>\$ORACLE_HOME/ldap/log/oidstack instance_Identifier dispatcher server PID.log</code> Note: The <code>oidstack.log</code> files pertain to SIGSEGV/SIGBUS tracing. Also, empty files of this name are created during directory instance startup, and can be ignored.
LDAP dispatcher (oidldapd)	<code>\$ORACLE_HOME/ldap/log/oidldapdXX.log</code> where <i>XX</i> is the server instance number
OID Monitor (oidmon)	<code>\$ORACLE_HOME/ldap/log/oidmon.log</code>

Table 3–2 (Cont.) Log File Locations

Component	Log File Name
OPMN	<code>\$ORACLE_HOME/opmn/logs/opmn.log</code> <code>\$ORACLE_HOME/opmn/logs/OID</code> <code>\$ORACLE_HOME/opmn/logs/OIDCTL</code> Log files for other Oracle Application Server components invoked by OPMN in <code>\$ORACLE_HOME/logs/opmn/</code>.
Replication setup (ldaprepl.sh)	<code>\$ORACLE_HOME/ldap/admin/LOGS/ldaprepl.log</code>

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](#)
- [Process Control of Oracle Internet Directory Components](#)
- [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: *Oracle Identity Management Guide to Delegated Administration*

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](#)

- [Configuring the Display and Duration of Searches in Oracle Directory Manager](#)
- [Performing Administrative 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. If no server instance is running, start one as described in ["Semantics of OPMN Starting Oracle Internet Directory"](#) on page 4-9.

See Also: ["Oracle Internet Directory Architecture"](#) on page 2-1 for a conceptual explanation of directory server instances

To start Oracle Directory Manager, follow the instructions for your operating system as described in [Table 4-1](#):

Table 4-1 *Operating System-Specific Instructions for Starting Oracle Directory Manager*

Operating System	Instructions
Microsoft Windows	From the Start menu, choose Programs , then <i>ORACLE_HOME</i> , then Integrated Management , then Oracle Directory Manager
UNIX	If you have not set the path, then navigate to <i>ORACLE_HOME/bin</i> . At the system prompt, enter: <code>oidadmin</code>

The first time you start Oracle Directory Manager, an alert tells you that you must connect to a server. Choose **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. These fields are described in [Table A-1](#) on page A-2.

See Also:

- [Chapter 13, "Secure Sockets Layer \(SSL\) and the Directory"](#) for instructions on enabling SSL
 - ["Entries"](#) on page 2-7 for instructions on formatting distinguished names
 - ["Configuring SSL Parameters"](#) on page 13-2 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. In the SSL tab page, enter the requested data in the fields. These fields are described in [Table A-2](#) on page A-3.
 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 4-2](#) 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 4–2 Oracle Directory Manager Menu Bar

Menu	Menu Items
File	<p>Create—Adds an object</p> <p>Create Like—Adds a new object by using the object selected in the navigator pane as a template</p> <p>Connect—Connects to a directory server selected in the navigator pane</p> <p>Disconnect—Disconnects from a directory server selected in the navigator pane</p> <p>Exit—Exits Oracle Directory Manager</p>
Edit	<p>Edit—Modifies an object</p> <p>Remove—Removes a selected object</p> <p>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 select Oracle Internet Directory, then <i>directory server instance</i>, then Server Management, then Object Classes, then this menu item searches for an object class. If you navigate to Oracle Internet Directory, then <i>directory server instance</i>, then Server Management, then Attributes, then it searches for attributes.</p>
View	<p>Refresh—Updates data stored in memory to reflect changes in the database</p> <p>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.</p>
Operation	<p>Create Object Class—Displays the New Object Class dialog box that you use to add a new object class</p> <p>Create Attribute—Displays the New Attribute Type dialog box that you use to add a new attribute to an entry</p> <p>Create Access Cutler Point—Displays the New Access Control Point dialog box that you use to add a new access control policy point (ACP).</p> <p>Create Entry—Displays the New Entry dialog box that you use to add a new directory entry</p> <p>Refresh Entry—Updates data for entries stored in memory to reflect changes in the database</p> <p>Refresh Subtree Entries—Updates the children of entries stored in memory to reflect changes in the database</p> <p>Configure Search Filter—Narrows the range of entries the navigator pane displays according to whatever filter you specify</p> <p>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.</p> <p>Search—Enables you to configure ACP searches</p> <p>User Preferences—Displays a dialog box that enables you to:</p> <ul style="list-style-type: none"> ■ Configure the display of entry search results ■ Establish whether ACPs are displayed whenever Oracle Directory Manager runs, or only as the result of a search
Help	<p>Contents—Displays the Contents tab page of the Help navigator</p> <p>Search for Help On...—Displays the Help Search dialog box that you use to search for words in the online help guide</p> <p>About Oracle Internet Directory—Displays Oracle Internet Directory version information</p>

The Oracle Directory Manager Toolbar

Table 4-3 illustrates and describes the buttons in the Oracle Internet Directory toolbar. Buttons become enabled or disabled depending on the pane or tab page you are displaying in Oracle Directory Manager.

Table 4-3 Oracle Directory Manager Toolbar

Button	Purpose
	Connect/Disconnect —Connects to or disconnect from a directory server selected in the navigator pane
	Refresh —Updates data for objects other than entries that are stored in memory to reflect changes in the database
	Create —Adds a new object
	Create Like —Adds a new object by using another object as a template
	Edit —Modifies an object
	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 > <i>directory server instance</i> > Server Management > Object Classes, then this button searches for an object class. If you navigate to Oracle Internet Directory > <i>directory server instance</i> > Server Management > Attributes, then it searches for attributes.
	Delete —Removes an object
	Add Object Classes —Adds an object class to an existing 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 click this button, an alert asks you to confirm that you want to drop the index.
	Search —Enables you to configure ACP searches
	User Preferences —Enables you to configure the display of ACPs in the navigator pane, as well as entries in a search operation
	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 earlier in this chapter, in "[Connecting to a Directory Server by Using Oracle Directory Manager](#)" on page 4-2.

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 Administrative Tasks by Using Oracle Directory Manager

You can perform most of the Oracle Internet Directory administrative tasks through Oracle Directory Manager. Those 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
- ["Oracle Identity Management Command-line Tool Reference"](#) in *Oracle Identity Management User Reference*

[Table 4–4](#) lists the task areas you can manage by using Oracle Directory Manager and where to find instructions for each area.

Table 4–4 Task Areas in Oracle Directory Manager

Task Area	Instructions
Access Control Management	"Managing Access Control by Using Oracle Directory Manager" on page 14-13 Managing Access Control by Using Command-Line Tools on page 14-34
Attribute Uniqueness Management	Chapter 7, "Attribute Uniqueness in the Directory"
Audit Log Management	Chapter 10, "Logging, Auditing, and Monitoring the Directory"
Change Log Management	"Change Logs in Directory Replication" on page 24-15 Chapter 25, "Oracle Internet Directory Replication Administration" The chapter on the Oracle Directory Synchronization Service in <i>Oracle Identity Management Integration Guide</i> The chapter on the Oracle directory integration and provisioning server in <i>Oracle Identity Management Integration Guide</i>
Entry Management	"Managing Entries by Using Oracle Directory Manager" on page 6-1
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 Internet Directory Replication Administration"

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

Task Area	Instructions
Schema Management	"Object Classes in the Directory" on page 8-1 "Attributes in the Directory" on page 8-8
Server Management	Chapter 5, "Oracle Directory Server Administration"

Process Control of Oracle Internet Directory Components

This section enumerates the concepts behind the process control model in Oracle Internet Directory. This applies to Oracle Internet Directory LDAP, Replication, and Directory Integration Server instances.

This section contains these topics:

- [Tools and Daemons Important to Oracle Internet Directory Process Control](#)
- [Oracle Internet Directory Integration with OPMN](#)
- [Oracle Internet Directory Process Control–Best Practices](#)
- [OIDMON and the ODS_PROCESS Table](#)
- [OIDCTL Process Control Semantics](#)

For information on starting and stopping the Oracle Directory Integration and Provisioning server, see the chapter on managing the Oracle Directory Integration and Provisioning server in *Oracle Identity Management Integration Guide*.

Tools and Daemons Important to Oracle Internet Directory Process Control

OPMN is a daemon process that monitors the different components in an installation of Oracle Application Server. It is installed and configured in every middle-tier and Oracle Application Server Infrastructure installation and is essential for running Oracle Application Server. Since Oracle Internet Directory is installed as part of the Oracle Application Server Infrastructure, OPMN is responsible for monitoring Oracle Internet Directory as an Oracle Application Server component. The command-line interface to OPMN is `$ORACLE_HOME/opmn/bin/opmnctl`. You use OPMNCTL to stop or start Oracle Internet Directory as a component.

See Also: The chapter entitled "Configuring Oracle Internet Directory" in *Oracle Process Manager and Notification Server Administrator's Guide*

OIDMON (`$ORACLE_HOME/bin/oidmon`) is a daemon process responsible for the process control of all Oracle Internet Directory Server instances.

OIDCTL (`$ORACLE_HOME/bin/oidctl`) is a command line tool that lets you configure additional Oracle Internet Directory Server Instances or perform process control at the instance level.

Oracle Internet Directory Integration with OPMN

This section describes Oracle Internet Directory interaction with OPMN. It includes the following sections:

- [Semantics of OPMN Monitoring Oracle Internet Directory](#)
- [Oracle Internet Directory Snippet in OPMN.XML](#)

- [Semantics of OPMN Starting Oracle Internet Directory](#)
- [Semantics of OPMN Stopping Oracle Internet Directory](#)
- [Semantics of OPMN Monitoring OIDMON](#)

Semantics of OPMN Monitoring Oracle Internet Directory

Monitoring rules are as follows:

- OPMN is responsible for monitoring Oracle Internet Directory as an Oracle Application Server component.
- OPMN integration with Oracle Internet Directory is such that OPMN knows only about OIDMON and is unaware of the Oracle Internet Directory Server Instances.
- OPMN is responsible for the direct start, stop, restart and monitoring of OIDMON only. OIDMON continues to be responsible for the direct start, stop, restart and monitoring of all Oracle Internet Directory Server Instances.

Oracle Internet Directory Snippet in OPMN.XML

Oracle Internet Directory component-specific directives are located as follows:

- Oracle Internet Directory component-specific directives for OPMN are located under the tag `<ias-component id="OID" status="enabled">` in `$ORACLE_HOME/opmn/conf/opmn.xml`.
- OPMN uses the directives in the OID component snippet in `opmn.xml` and invokes OIDMON and OIDCTL as required.
- OIDCTL related requirements are located under the tag `<category id="oidctl parameters">`.
- OIDMON related requirements are located under the tag `<category id="oidmon parameters">`. There should be only one such directive.
- The default value of OID Snippet in `opmn.xml` has one entry for OIDMON and one for OIDCTL.

See Also: The chapter entitled "Configuring Oracle Internet Directory" in *Oracle Process Manager and Notification Server Administrator's Guide*

Semantics of OPMN Starting Oracle Internet Directory

OPMN startup of an Oracle Internet Directory component proceeds as follows:

- You can indicate to OPMN the intent to start an Oracle Internet Directory component with one of the following commands:


```
opmnctl startall
```

```
opmnctl startproc ias-component=OID
```
- OPMN issues an `oidmon start` command with appropriate arguments to OIDMON as specified in the contents of "oidmon parameters" in the OID Snippet in `opmn.xml`.
- OPMN issues `oidctl start` commands if the OID Snippet in `opmn.xml` has entries that require this.

Note that when you use `opmnctl startproc ias-component=OID`, the `opmn.xml` parameters are not reloaded as they are when you use `opmnctl startall`

Semantics of OPMN Stopping Oracle Internet Directory

OPMN stopping of an Oracle Internet Directory component proceeds as follows:

- You can indicate to OPMN the intent to stop an Oracle Internet Directory component with one of the following commands:

```
opmnctl stopall
```

```
opmnctl stopproc ias-component=OID
```

- OPMN issues an `oidmon stop`.
- OPMN does **not** issue `oidctl stop` commands; instead, the OIDMON stop semantics ensure that the Oracle Internet Directory Server Instances are stopped as required. For more information, see "[OIDMON and the ODS_PROCESS Table](#)" on page 4-13.

Note that when you use `opmnctl stopproc ias-component=OID`, the `opmn.xml` parameters are not reloaded as they are when you use `opmnctl stopall`

Semantics of OPMN Monitoring OIDMON

OPMN monitors OIDMON as follows:

- Once you start OIDMON through OPMN, OPMN ensures that OIDMON is up and running. If OIDMON goes down for some reason, OPMN will bring it back up.
- If you issue `oidmon stop` on the command line, OIDMON will be stopped but OPMN will immediately bring it back up.

Oracle Internet Directory Process Control–Best Practices

The recommended approach for using OPMNCTL and OIDCTL is as follows:

- Use OPMNCTL to stop or start Oracle Internet Directory as a component. That is, use it to stop or start all Oracle Internet Directory LDAP, replication, and Oracle Directory Integration and Provisioning server instances.
 - Using OPMNCTL to stop Oracle Internet Directory causes OPMN to issue an `oidmon stop`, which results in OIDMON shutting down all configured LDAP, replication, and Oracle Directory Integration and Provisioning server instances.
 - Using OPMNCTL to start Oracle Internet Directory causes OPMN to issue an `oidmon start`, which results in OIDMON starting up all configured LDAP, replication, and Oracle Directory Integration and Provisioning server instances.
- Use OIDCTL to configure required additional Oracle Internet Directory Server Instances.
 - To configure an Oracle Internet Directory LDAP, replication, or Oracle Directory Integration and Provisioning server that is not part of the default configuration, use the OIDCTL command to start such an instance.

- Issue the `oidctl start` command only once for each instance for the duration of the deployment of that configuration. The OIDMON start and stop semantics ensure that configured servers start and stop appropriately.
- Use OIDCTL to perform process control at the instance level only.
 - Use `oidctl stop` to stop a particular instance of the Oracle Internet Directory LDAP, replication, or Oracle Directory Integration and Provisioning server. Do not use it to stop all the configured instances of the servers.
 - Use `oidctl start` to start an additional instance of the Oracle Internet DirectoryLDAP, replication, or Oracle Directory Integration and Provisioning server that is not already configured.
 - If, for some reason, a server instance is stopped using `oidctl stop`, then it is removed from the process table and will not be known to OIDMON any longer. To restart an instance that was stopped this way, use `oidctl start`.

The following sections provide examples of the recommended approach. The examples are:

- [Changing the Configuration of the Default OID LDAP Server Instance](#)
- [Configuring Additional Oracle Internet Directory LDAP Server Instances](#)
- [Deconfiguring the Default Oracle Internet Directory LDAP Server Instance](#)
- [Configuring an Instance of the Oracle Internet Directory Replication Server](#)
- [Configuring an Oracle Directory Integration and Provisioning Server Instance](#)

See Also: "Oracle Internet Directory Server Administration Tools" in *Oracle Identity Management User Reference* for more information on the syntax of the commands used in the examples.

Changing the Configuration of the Default OID LDAP Server Instance

A default Oracle Internet Directory installation uses the default configuration set (`configset0`), which provides a single server instance with one server process and two database connections configured. This configuration might be inadequate to handle the production LDAP load in your environment. If so, you need to increase the number of server processes or database connections or both. You change these by changing the `orclserverprocs` and `orclmaxcc` attributes values, respectively, in `configset0`.

You change the default `configset` using Oracle Directory Manager, as follows:

1. Launch Oracle Directory Manager.
2. Log in as `orcladmin`.
3. Expand **Server Management**.
4. Click **Default Configuration**.
5. Change **Max Number of DB Connections** to the desired value. A typical recommendation is 10.
6. Change **Number of Child Process** if required. A typical value is 1 or the number of CPUs on the system.
7. Click **Apply**.
8. Restart the Oracle Internet Directory servers, as follows:

```
opmnctl stopproc ias-component=OID
```

```
opmnctl startproc ias-component=OID
```

Oracle recommends that you not change other parameters in the default configuration set.

Configuring Additional Oracle Internet Directory LDAP Server Instances

To start additional Oracle Internet Directory LDAP server instances, add additional configuration sets with the required configuration values and use these additional configuration sets to start additional server instances. (Do not use the default configuration set and override the default values.) You add a configuration set and to use it to start an Oracle Internet Directory LDAP Server Instance as follows:

1. Launch Oracle Directory Manager.
2. Expand **Server Management**.
3. Expand **Directory Server**.
4. Right click **Default Configuration Set**.
5. Click **Create Like**.
6. Change the required parameters in the new configuration set. Ensure that the port numbers do not overlap with those of the default configuration set or any other configuration set.
7. Click **OK**.

To start an LDAP server instance using a new configuration set called `configset2`, type:

```
oidctl connect=connStr server=oidldapd instance=2 configset=2 start
```

Deconfiguring the Default Oracle Internet Directory LDAP Server Instance

To replace the Oracle Internet Directory LDAP server instance with one or more Oracle Internet Directory LDAP Server Instances, you must edit `opmn.xml` to deconfigure the default LDAP instance. By default, `opmn.xml` contains an XML snippet that attempts to start the default Oracle Internet Directory LDAP server instance when you type `opmnctl start`. To deconfigure the default Oracle Internet Directory LDAP server instance, perform the following steps:

1. Type:

```
oidctl connect=connStr server=oidldapd instance=1 stop
```

2. Edit the file `$ORACLE_HOME/opmn/conf/opmn.xml` and remove the following lines:

```
<category id="oidctl-parameters">  
<data id="connect" value="iasdb"/>  
<data id="startoidldapd" value="true"/>  
</category>
```

Configuring an Instance of the Oracle Internet Directory Replication Server

To configure an instance of OID Replication Server, use the `oidctl start` command. For example:

```
oidctl connect=connStr server=oidrepld instance=1 \  
flags="-h LdapHost -p LdapPort" start
```

Do not start more than one instance of `oidrepld`.

Configuring an Oracle Directory Integration and Provisioning Server Instance

To configure an instance of the Oracle Directory Integration and Provisioning Server, use the `oidctl start` command. For example:

```
oidctl connect=connStr server=odisrv instance=1 \
  flags="-h LdapHost -p LdapPort" start
```

OIDMON and the ODS_PROCESS Table

OIDMON is responsible for starting, stopping, restarting, and monitoring of all the Oracle Internet Directory Server instances including Oracle Internet Directory LDAP, Replication, and Directory Server instances.

OIDMON reads the contents of the `ods_process` table in the ODS database user schema once every periodicity and acts upon the intent conveyed by the contents of that table. The periodicity is controlled by the value of the `sleep` command line argument used at `oidmon` startup, and the default value is 10 seconds.

Table 4-5 describes the information in the `ODS_PROCESS` table that is relevant to process control:

Table 4-5 Process Control Items in the ODS_PROCESS Table

Item	Meaning
Instance	Unique instance number for a given server ID on a given host
PID	Process ID of the server that is up and running
ServerID	Server ID (2=OIDLDAPD, 3=OIDREPLD, 7=ODISRV)
Flags	Command line arguments that need to be passed to the server instance
Hostname	Name of the host on which this server must be present
Configset	Configset information
State	State of the Server Instance (0=stop, 1=start, 2=running, 3=restart, 4=shutdown)
RetryCount	Number of attempts to start the server instance before it could be started successfully

Notes:

- There is a uniqueness constraint on (Instance,ServerID,Hostname).
- Details are provided here about `ods_process` only to convey the concepts. Any updates to the table by a user, other than by using `OIDCTL`, are inappropriate and are not supported by Oracle.

OIDMON takes the following actions with respect to Oracle Internet Directory server instances:

- When OIDMON is stopped, it performs the following tasks before shutting down:
 - OIDMON stops all the server instances active (up and running) on its node, that is, all instances active on the same host as OIDMON

- OIDMON updates the value of the "state" column of rows in the ods_process table with matching hostname to 4
- When OIDMON is started, it starts all Oracle Internet Directory Server instances whose information in the ods_process table has state value 1 or 4 and hostname value matching the host on which OIDMON is active.

OIDCTL Process Control Semantics

This section explains the semantics of starting and stopping Oracle Internet Directory server instances by using OIDCTL.

OIDCTL communicates the intent to start, stop, or restart a particular Oracle Internet Directory Server instance by updating the ods_process table in the ODS database user schema.

When you start an Oracle Internet Directory server instance:

- You use the OIDCTL command line utility to indicate the intent to start a particular Oracle Internet Directory server instance
- OIDCTL inserts a row into the ods_process table to convey the intent to OIDMON
- If the uniqueness constraint on the ods_process table is violated, then OIDCTL reports the error "Instance number already in use"
- OIDMON reads this information, starts the server instance, and updates the state and PID columns of the ods_process table as appropriate

When you stop an Oracle Internet Directory server instance by using OIDCTL:

- You use the OIDCTL command line utility to indicate the intent to stop a particular Oracle Internet Directory server instance
- OIDCTL updates the corresponding row in the ods_process table to convey the intent to OIDMON.
- If the corresponding row is not found, that is, the given instance is not configured, then OIDCTL reports the error "Cannot Stop an Instance that is not running"
- OIDCTL updates the state value to 0
- OIDMON reads this information, stops the server instance, and deletes the row representing this server instance from the ods_process table

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 indexing existing attributes
- 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.
-
-

Note: Command-line examples in Oracle Identity Management documentation are based on the UNIX `ksh`. Arguments that must be escaped from the shell are shown in double quotes (`"`). Use the appropriate quote characters for your shell environment.

See Also: "LDIF File Formatting Rules" in *Oracle Identity Management User Reference* for information on formatting an LDIF file

This section contains these topics:

- [Command-Line Tools for Starting, Stopping, and Monitoring Oracle Internet Directory Servers](#)
- [Command-Line Tools for Managing Entries and Attributes](#)
- [Command-Line Tools for Performing Bulk Operations](#)
- [Command-Line Tools for Managing Replication](#)
- [OID Migration Tool \(Idifmigrator\)](#)
- [OID Database Statistics Tool \(oidstats.sql\)](#)
- [OID Database Password Utility \(oidpasswd\)](#)

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
Oracle Process Manager and Notification Server (OPMN)	Use OPMNCTL to stop or start Oracle Internet Directory as a component of Oracle Application Server.	The "opmnctl" command-line tool reference in <i>Oracle Identity Management User Reference</i> . The chapter entitled "Configuring Oracle Internet Directory" in <i>Oracle Process Manager and Notification Server Administrator's Guide</i>
OID Control Utility (OIDCTL)	Use this tool to start and stop an individual instance of the server. The commands are interpreted and executed by the OID Monitor process.	" Oracle Internet Directory Architecture " on page 2-1 for a conceptual description The "oidctl" command-line tool reference in Oracle Identity Management User Reference
OID Monitor (OIDMON)	You do not need to invoke OIDMON directly. You start and stop it using OPMN. When you issue commands through OID Control Utility (OIDCTL) to start or stop directory server instances, your commands are interpreted by OIDMON.	" Oracle Internet Directory Architecture " on page 2-1 for a conceptual description The "oidmon" command-line tool reference in Oracle Identity Management User Reference 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.sh" command-line tool reference in <i>Oracle Identity Management User Reference</i> for syntax and usage notes " Indexing an Attribute by Using Oracle Directory Manager " on page 8-12 " Indexing an Attribute by Using Oracle Directory Manager " on page 8-12
ldapadd	Use this tool to add entries one at a time.	The "ldapadd" command-line tool reference in <i>Oracle Identity Management User Reference</i>
ldapaddmt	Use this tool to add several entries concurrently by using this shared-server tool.	The "ldapaddmt" command-line tool reference in <i>Oracle Identity Management User Reference</i>
ldapbind	Use this tool to authenticate user/client to a directory server.	The "ldapbind" command-line tool reference in <i>Oracle Identity Management User Reference</i>

Table 4–7 (Cont.) Tools for Managing Entries

Tool	Description	More Information
ldapcompare	Use this tool to see whether an entry contains a specified attribute value.	The "ldapcompare" command-line tool reference in <i>Oracle Identity Management User Reference</i>
ldapdelete	Use this tool to delete entries.	The "ldapdelete" command-line tool reference in <i>Oracle Identity Management User Reference</i>
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.	The "ldapmoddn" command-line tool reference in <i>Oracle Identity Management User Reference</i>
ldapmodify	Use this tool to create, update, and delete attribute data for an entry.	The "ldapmodify" command-line tool reference in <i>Oracle Identity Management User Reference</i>
ldapmodifymt	Use this tool to modify several entries concurrently by using this shared-server tool.	The "ldapmodifymt" command-line tool reference in <i>Oracle Identity Management User Reference</i>
ldapsearch	Use this tool to search for directory entries.	The "ldapsearch" command-line tool reference in <i>Oracle Identity Management User Reference</i>

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	The "bulkdelete" command-line tool reference in <i>Oracle Identity Management User Reference</i>
bulkload	Use this tool to load and append large numbers of entries to Oracle Internet Directory through LDIF files	The "bulkload" command-line tool reference in <i>Oracle Identity Management User Reference</i>
bulkmodify	Use this tool to modify a large number of existing entries efficiently	The "bulkmodify" command-line tool reference in <i>Oracle Identity Management User Reference</i>
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.	The "ldifwrite" command-line tool reference in <i>Oracle Identity Management User Reference</i>

Command-Line Tools for Managing Replication

[Table 4–9](#) on page 4-18 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 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 "remtool" command-line tool reference in <i>Oracle Identity Management User Reference</i> 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: <ol style="list-style-type: none"> 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 Oracle Internet Directory Reconciliation Tool " on page 25-19 The "oidreconcile" command-line tool reference in <i>Oracle Identity Management User Reference</i>
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 25-19 The "hiqretry" command-line tool reference in <i>Oracle Identity Management User Reference</i> for syntax and an explanation of how OID Reconciliation Tool works

OID Migration Tool (Idifmigrator)

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

See Also: The "Idifmigrator" command-line tool reference in *Oracle Identity Management User Reference* for instructions on using this tool

OID Database Statistics Tool (oidstats.sql)

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: The "oidstats.sql" command-line tool reference in *Oracle Identity Management User Reference*

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 `oidpwdlldap1`, for the Oracle Internet Directory database password, and a wallet, named `oidpwrssid`, for the Oracle directory replication server password.

The `ssid` 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`.

Note: To change the ODS database user password, you must use the `oidpasswd` tool. If you change the ODS database user password by any other means, then Oracle Internet Directory instances fail to start.

- Reset the super user password
- Manage super user restricted ACPs

See Also: The "oidpasswd" command-line tool reference in *Oracle Identity Management User Reference*

Routine Administration at a Glance

Oracle Internet Directory routine administration tasks are described throughout this manual. [Table 4-10](#) points you to the information you need for some of the more common tasks.

Table 4-10 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 8-13
Add, modify, or delete an attribute by using the Oracle Directory Manager	"Attributes in the Directory" on page 8-8
Managing Entries	-
Add, modify, or delete a directory entry by using command-line tools	"Managing Entries by Using Command-Line Tools" on page 6-7
Add, modify, or delete a directory entry by using Oracle Directory Manager	"Managing Entries by Using Oracle Directory Manager" on page 6-1

Table 4–10 (Cont.) Routine Administration Tasks

Task	Information
Import bulk data files	The "bulkload" command-line tool reference in <i>Oracle Identity Management User Reference</i> "LDIF File Formatting Rules and Examples" in <i>Oracle Identity Management User Reference</i>
View Directory Information Tree (DIT) hierarchy of entries	"Managing Entries by Using Oracle Directory Manager" on page 6-1
Managing Object Classes	-
Add, modify, or delete object classes by using command-line tools	"Managing Object Classes by Using Command-Line Tools" on page 8-7
Add, modify, or delete object classes by using Oracle Directory Manager	"Object Classes in the Directory" on page 8-1
Managing Replication	-
Set up replication	Chapter 25, "Oracle Internet Directory Replication Administration"
Resolve replication change conflicts	"Resolving Conflicts Manually in a Multimaster Replication Group" on page 25-18
Move replication changes from human intervention queue to either the retry queue or the purge queue	"About the Oracle Internet Directory Reconciliation Tool" on page 25-19
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 5-5
Configure server instance parameters by using Oracle Directory Manager	"Managing Server Configuration Set Entries by Using Oracle Directory Manager" on page 5-3
Connect to a directory by using Oracle Directory Manager	"Connecting to a Directory Server by Using Oracle Directory Manager" on page 4-2 "Connecting to Additional Directory Servers by Using Oracle Directory Manager" on page 4-5
Start the directory server processes	Chapter 3, "Post-Installation Tasks and Information"
Stop the directory server processes	Chapter 3, "Post-Installation Tasks and Information"
View system operational attributes	"Setting System Operational Attributes by Using Oracle Directory Manager" on page 5-7

Part II

Basic Directory Administration

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

- [Chapter 5, "Oracle Directory Server Administration"](#)
- [Chapter 6, "Directory Entries Administration"](#)
- [Chapter 7, "Attribute Uniqueness in the Directory"](#)
- [Chapter 8, "Directory Schema Administration"](#)
- [Chapter 9, "Dynamic and Static Groups in Oracle Internet Directory"](#)
- [Chapter 10, "Logging, Auditing, and Monitoring the Directory"](#)
- [Chapter 11, "Backup and Restoration of a Directory"](#)

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](#)
- [Managing Anonymous Binds](#)
- [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, "Post-Installation 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 **object class**, 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 2-6 for a conceptual overview of configuration set entries

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

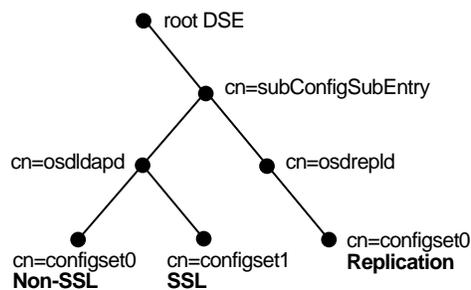


Figure 5-1 shows:

- An Oracle directory server (`cn=odsldapd`) 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=odsrepld`) 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 Internet Directory Replication Administration"](#) for information about configuration parameters for replication
- "OID Configuration Schema Elements" in *Oracle Identity Management User Reference* 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.

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 "Oracle Internet Directory Server Administration Tools" in *Oracle Identity Management User Reference*.

This section contains the following topics:

- [Viewing Configuration Set Entries by Using Oracle Directory Manager](#)
- [Adding Configuration Set Entries by Using Oracle Directory Manager](#)
- [Modifying Configuration Set Entries by Using Oracle Directory Manager](#)
- [Deleting Configuration Set Entries by Using Oracle Directory Manager](#)

Viewing Configuration Set Entries by Using Oracle Directory Manager

To view configuration set entries:

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then ***directory server instance***, then **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 5-5

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 **Oracle Internet Directory Servers**, then *directory server instance*, then **Server Management**, then **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 A-34](#) on page A-21.
5. Select the **SSL Settings** tab and fill in the fields. These are described in [Table A-35](#) on page A-21.
6. Choose **Apply**.
7. Restart the server instance for the command to take effect.

See Also:

- The "oidctl" command-line tool reference in *Oracle Identity Management User Reference*
- *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 10-4

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

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then *directory server instance*, then **Server Management**, then **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 A-34](#) on page A-21.
5. Select the **SSL Settings** tab and fill in the fields. These are described in [Table A-35](#) on page A-21.
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 A-34](#) on page A-21. To save the changes, choose **Apply**.
4. Select the **SSL Settings** tab and modify the fields. These are described in [Table A-35](#) on page A-21. To save the changes, choose **Apply**.
5. Restart the server instance for the command to take effect.

See Also:

- The "oidctl" command-line tool reference in *Oracle Identity Management User Reference*
- *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: The "oidctl" command-line tool reference in *Oracle Identity Management User Reference*

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 **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: "LDIF File Formatting Rules and Examples" in *Oracle Identity Management User Reference* for information on LDIF

This section contains the following topics:

- [Adding Configuration Set Entries by Using ldapadd](#)

- [Modifying and Deleting Configuration Set Entries by Using ldapmodify](#)

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:

- "LDIF File Formatting Rules and Examples" in *Oracle Identity Management User Reference*
- The "ldapadd" command-line tool reference in *Oracle Identity Management User Reference* for a detailed list of options available with this command
- "OID Configuration Schema Elements" in *Oracle Identity Management User Reference* 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:

- "LDIF File Formatting Rules and Examples" in *Oracle Identity Management User Reference*
- The "ldapmodify" command-line tool reference in *Oracle Identity Management User Reference* for a more detailed discussion of ldapmodify, and a list of its options
- "OID Configuration Schema Elements" in *Oracle Identity Management User Reference* 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 2-9

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 A-36](#) on page A-22 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. The "Oracle Identity Management LDAP Attribute Reference" in *Oracle Identity Management User Reference* describes the modifiable system operational attributes.

See Also: The "ldapmodify" command-line tool reference in *Oracle Identity Management User Reference* 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.

You can use either Oracle Directory Manager or ldapmodify to publish a naming context. 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

This section contains these topics:

- [About Super Users, Guest, Users, and Proxy Users](#)

- [Managing Super Users, Guest Users, and Proxy Users by Using Oracle Directory Manager](#)
- [Managing Super Users, Guest Users, and Proxy Users by Using ldapmodify](#)

About Super Users, Guest Users, and Proxy Users

[Table 5–1](#) on page 5-9 defines super users, guest users, and proxy users.

You can use either Oracle Directory Manager or ldapmodify to administer the user names and passwords for each of these users.

Table 5–1 *Definitions of Super User, Guest User, and Proxy User*

Type of User	Definition
Super user	A special directory administrator with full access to directory information. The default user name of the super user is <code>orcladmin</code> ; the default password is <code>welcome</code> . Note: Oracle recommends that you change the password immediately after installation.
Guest user	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 <code>guest</code> ; the default password is <code>guest</code> .
Proxy user	A user typically 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 <code>proxy</code> ; the default password is <code>proxy</code> . See Also: " Indirect Authentication " on page 12-4 for more information about proxy users

See Also: [Chapter 14, "Directory Access Control"](#) for information on how to set access rights

Note: You can 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.

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 A–37](#) on page A-25. 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 5–2 Names, Passwords, and Attributes for Super, Guest, and Proxy Users

User Name	Password	Attribute
<i>Super user name</i>	orclsupassword	orclsuname
<i>Guest user name</i>	orclgupassword	orclguname
<i>Proxy user name</i>	orclprpassword	orclprname

For example, to change the password of the super user to `superuserpassword`, use ldapmodify to modify the [DSE](#) by using an LDIF file containing the following:

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

See Also: The "ldapmodify" command-line tool reference in *Oracle Identity Management User Reference* for ldapmodify syntax and usage notes.

Managing Anonymous Binds

The Oracle Directory Server can be configured to allow or disallow anonymous bind. This behavior is controlled by the `orclAnonymousBindsFlag` attribute of the root DSE entry. [Table 5–3](#) lists the allowed values for `orclAnonymousBindsFlag` and the resulting directory server behavior.

Table 5–3 Orclanonymousbindflag Value and Directory Server Behavior

orclAnonymousBindsFlag Value	Directory Server Behavior
0	Disallows anonymous bind
1	Allows anonymous bind (default)
2	Allows anonymous bind but allows only search operations on root DSE entry for anonymous users

By default, the directory server allows anonymous bind. That is, the `orclAnonymousBindsFlag` attribute is set to 1.

The following example shows how to disable anonymous bind using the `ldapmodify` command-line tool:

```
ldapmodify -h hostname -p port -D cn=orcladmin -w super_user_pwd <<EOF
dn:
changetype: modify
replace: orclanonymousbindsflag
orclanonymousbindsflag: 0
EOF
```

See Also:

- ["Setting System Operational Attributes"](#) on page 5-7
- [Table A-36, " System Operation Attributes Displayed in Oracle Directory Manager"](#) on page A-22

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 5-3 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 "Oracle Identity Management LDAP Attribute Reference" in *Oracle Identity Management User Reference*.

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: The "oidpasswd" command-line tool reference in *Oracle Identity Management User Reference*

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=myCompany,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:

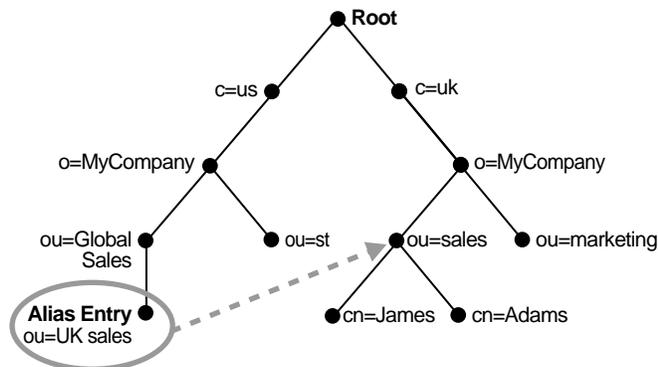
```
(2.5.6.1 NAME 'alias' SUP top STRUCTURAL MUST aliasedObjectName)
```

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 distinguishedNameMatch 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: 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 5-13.

1. Create a sample LDIF file, `My_file.ldif`, with the following entries:

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

dn: o=MyCompany, c=us
o: MyCompany
objectclass: organization

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

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

dn: cn=President, o=MyCompany, c=us
objectclass: alias
aliasedobjectname: cn=John Doe, o=MyCompany, c=us
```

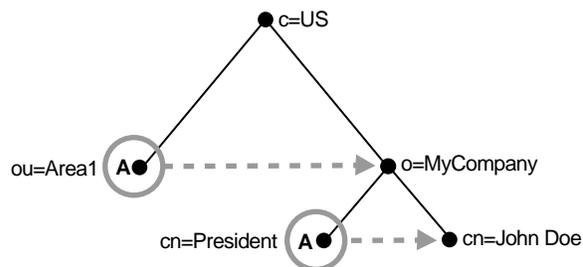
2. Add these entries to the directory by using the following command:

```
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 5-15 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=Areal,c=us` with a filter of `"objectclass=*"` with the dereferencing flag set to `-a find`.

```
ldapsearch -p port -h host -b "ou=Areal,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=Areal,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=Areal,c=us"` with a filter of `"objectclass=*"` with the dereferencing flag set to `-a find`.

```
ldapsearch -p port -h host -b "ou=Areal,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=Areal,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=Areal,c=us"` with a filter of `"objectclass=*"` with the dereferencing flag set to `-a find`.

```
ldapsearch -p port -h host -b "ou=Areal,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=Areal,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

[Table 5-4](#) lists the messages related to alias entry dereferencing and the corresponding meaning for each message.

Table 5-4 Entry Alias Dereferencing Messages

Message	Meaning
Alias Problem	Either of the following have occurred: <ul style="list-style-type: none"> ■ 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.
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 <code>aliasedObjectName</code> 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* 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. The client looks for this file in the following file system directories, in order of precedence:

1. The directory pointed to by the `LDAP_ADMIN` environment variable
2. The directory `ORACLE_HOME/ldap/admin` (or, on Microsoft Windows, `ORACLE_HOME\ldap\admin`)
3. The directory pointed to by the `TNS_ADMIN` environment variable
4. The directory `ORACLE_HOME/network/admin` (or, on Microsoft Windows, `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

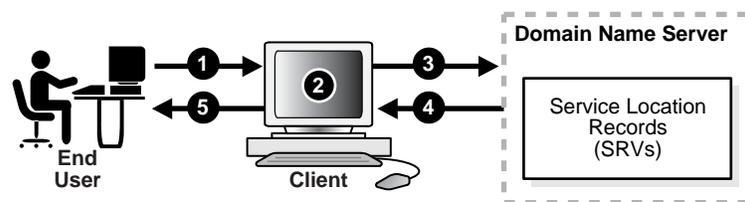
To find the directory server on which an entry resides, a client communicates with the domain name server. Specifically, it provides to the domain name server a domain name. The domain name specifies where the needed directory server is located.

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 5–5](#) 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 5-5 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 5-5 Arguments in a Service Location Record (SRV)

Argument	Description
Service	For a non-SSL-based server, the value for this argument is <code>ldap</code> . For an SSL-based server, the value is <code>ldaps</code> .
Proto	The value is always <code>tcp</code> .
Product	The value is always <code>oid</code> .
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 5-17
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 <code>SRV</code> .
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: <ol style="list-style-type: none"> 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. Compute the sum of the weights of those resource records, and with each resource record associate the running sum in the selected order. 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. Remove this SRV resource record from the set of the unordered SRV resource records. Apply the described algorithm to the unordered SRV resource records to select the next target host. Continue the ordering process until there are no unordered SRV resource records. Repeat this process for each priority.
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.

Directory Entries Administration

This chapter explains how to view, add, modify, and delete entries. It 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](#)
- [Viewing Attributes for a Specific Entry by Using Oracle Directory Manager](#)
- [Adding Entries by Using Oracle Directory Manager](#)
- [Modifying Entries by Using Oracle Directory Manager](#)
- [Managing Entries with Attribute Options by Using Oracle Directory Manager](#)

Searching for Entries by Using Oracle Directory Manager

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

To display an entry, in the navigator pane, expand **Oracle Internet Directory Servers**, then **directory server instance**, then **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 **Oracle Internet Directory Servers**, then **directory server instance**, then **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 A-39](#) on page A-27.
 - 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 A-40](#) on page A-27.
 8. Click **Search**. The results of your search appear in the **Distinguished Name** box.

See Also: "[Viewing Active Server Instance Information](#)" on page 5-11 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 6-1.

See Also: "[Viewing All Directory Attributes by Using Oracle Directory Manager](#)" on page 8-9 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 users 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.

Note: You must assign user entries to the `inetOrgPerson` object class in order for the entries to appear in the Oracle Internet Directory Self-Service Console.

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**.
3. In the right pane, the search interface appears. Use it to search for an entry that you want to use as a template.
4. From the entries retrieved, double-click one that you want to use as your template. The Entry dialog box for that entry appears.
5. In the Entry dialog box, click **Create Like**. A New Entry: Create Like dialog box appears.
6. 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.
7. 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.

Note: You cannot use a tilde (~) in a user name.

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 6-1
- ["Managing Group Entries"](#) on page 9-5
- ["Security Groups"](#) on page 14-3
- ["Globalization Support"](#) on page 2-15 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, then no special rules apply, and 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 6-1.
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 6-4.

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 6-1 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 10g Release 2 (10.1.2), Oracle Directory Manager does not allow you to add an attribute option to an entry when you create the entry. You can use Oracle Directory Manager to add attribute options only to already existing entries.

To add an attribute option to an existing entry:

1. 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](#)
- [Managing Entries with Attribute Options by Using Command-Line Tools](#)

Command-Line Tools for Managing Entries

[Table 6–1](#) lists each of the command-line tools for managing entries, and tells you where to find syntax and usage notes for each one.

Table 6–1 *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.	The "ldapadd" command-line tool reference in <i>Oracle Identity Management User Reference</i>
ldapaddmt	Add several entries concurrently by using this shared server tool.	The "ldapaddmt" command-line tool reference in <i>Oracle Identity Management User Reference</i>
ldapbind	Authenticate a user or client to a directory server. Verify that you can connect a client to a server.	The "ldapbind" command-line tool reference in <i>Oracle Identity Management User Reference</i>
ldapcompare	Compare attribute values you specify with those in a directory entry.	The "ldapcompare" command-line tool reference in <i>Oracle Identity Management User Reference</i>
ldapdelete	Delete entries.	The "ldapdelete" command-line tool reference in <i>Oracle Identity Management User Reference</i>
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.	The "ldapmoddn" command-line tool reference in <i>Oracle Identity Management User Reference</i>

Table 6–1 (Cont.) Command-Line Tools for Managing Entries

Tool	Task(s)	Syntax and Usage Notes
ldapmodify	Create, update, and delete attribute data for an entry. Modify configuration set entries. Modify DN or RDN of an entry.	The "ldapmodify" command-line tool reference in <i>Oracle Identity Management User Reference</i>
ldapmodifymt	Modify several entries concurrently by using this shared server tool.	The "ldapmodifymt" command-line tool reference in <i>Oracle Identity Management User Reference</i>
ldapsearch	Search for directory entries.	The "ldapsearch" command-line tool reference in <i>Oracle Identity Management User Reference</i>

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
- You cannot insert a tilde (~) in a user name.

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 2-11

Managing Entries by Using Bulk Tools

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

This section contains these topics:

- [Importing an LDIF File by Using bulkload](#)
- [Converting Directory Data to LDIF](#)
- [Modifying a Large Number of Entries](#)
- [Deleting a Large Number of Entries](#)

Note: If you do not use the bulkload utility to populate the directory, then you must run the oidstats.sql tool to avoid significant search performance degradation.

See Also:

- The "oidstats.sql" command-line tool reference in *Oracle Identity Management User Reference* for a description and syntax for the oidstats.sql 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:

- To run shell script tools on the Windows operating system, you need either of the following UNIX emulation utilities: Cygwin 1.3.2.2-1 or later (<http://sources.redhat.com>) or MKS Toolkit 8.6 or later. (Visit: <http://www.datafocus.com/>)
-
-

See Also: The "bulkload" command-line tool reference in *Oracle Identity Management User Reference*.

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 4: Generate the Input Files for SQL*Loader](#)
- [Task 5: Load the Input Files](#)
- [If Bulk Loading Fails](#)

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: *Oracle Database Backup and Recovery Basics* in the Oracle Database Documentation Library

Task 2: Find Out the Oracle Internet Directory Password

To use `bulkload` and the other shell script tools that have commands that end with `.sh`, you must provide the Oracle Internet Directory password. The default password is `ods`, although the system administrator can change it by using the [OID Database Password Utility](#).

See Also: The "oidpasswd" command-line tool reference in *Oracle Identity Management User Reference*

Task 3: Check Input for Schema and Data Consistency Violations

On UNIX, the `bulkload.sh` file usually resides in `ORACLE_HOME/ldap/bin`. On Microsoft Windows, 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: The "ldifwrite" command-line tool reference in *Oracle Identity Management User Reference*

Modifying a Large Number of Entries

The bulkmodify utility enables you to modify a large number of existing entries efficiently.

See Also: The "bulkmodify" command-line tool reference in *Oracle Identity Management User Reference*

Deleting a Large Number of Entries

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

See Also: The "bulkdelete" command-line tool reference in *Oracle Identity Management User Reference*

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** and the `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 2-18 for an overview of knowledge references and a description of a **smart knowledge reference** and a **default knowledge reference**

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=f00,c=uk` receives a referral.
- A client querying Server C with base `o=f00,c=us` receives a referral.
- An add operation of `o=f00,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 **directory 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 2-19 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
```

Attribute Uniqueness in the Directory

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

- [About Attribute Uniqueness](#)
- [Rules for Creating Attribute Uniqueness](#)
- [Managing Attribute Uniqueness](#)
- [Limitations of Attribute Uniqueness in Oracle Internet Directory 10g Release 2 \(10.1.2\)](#)

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 choose to enforce attribute uniqueness in SubscriberCompany1 only.
- Across one object class
For example, suppose that `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 you provide values for the attributes in [Table 7-1](#) on page 7-2.

Table 7-1 Attribute Uniqueness Constraint Entry

Attribute Name	Mandatory?	Valid Value	Default Value	Default Effect
orcluniqueattrname	Yes	Any string	N/A	N/A
orcluniquescopes	No	One of the following: <ul style="list-style-type: none"> ▪ base—Searches the root entry only ▪ onelevel—Searches one level only ▪ sub—Searches the entire directory 	sub	Searches the entire directory
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.

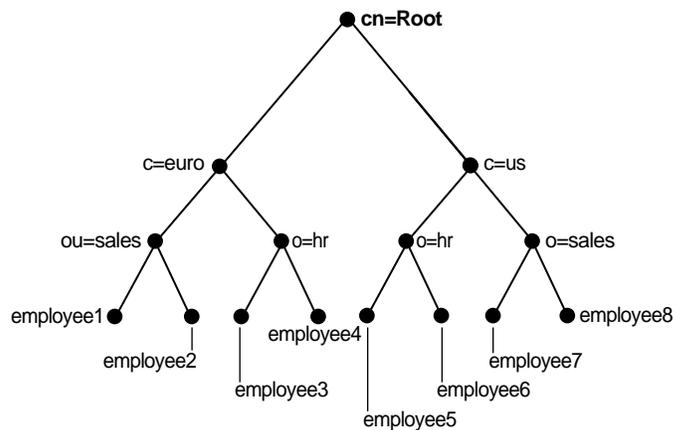
Note: The attribute uniqueness feature works on indexed attributes only.

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 7-1](#).

Figure 7-1 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 7-1](#) on page 7-3. 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 `employee3`, `employee4`, `employee7` and `employee8`.

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 `orcluniquescopes`, the attribute uniqueness constraint with the largest search scope takes effect.

For example, referring to [Figure 7-1](#) on page 7-3, suppose that a user defines two attribute uniqueness constraints as follows:

Constraint1:

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

Constraint2:

```
orcluniqueattrname: employee_id
orcluniquesubtree: c=us, cn=root
orcluniquescopes: 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 `orcluniquescopes`, but different values in `orcluniqueobjectclass`, then the union of attributes belonging to those object classes is checked.

For example, look at [Figure 7-1](#) on page 7-3. 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 7-1](#) on page 7-3, 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](#)
- [Managing Attribute Uniqueness by Using Oracle Directory Manager](#)
- [Managing Attribute Uniqueness by Using Command-Line Tools](#)

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 A-8](#) on page A-5.
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](#)
- [Creating Attribute Uniqueness Constraint Entries by Using Command-Line Tools](#)
- [Modifying Attribute Uniqueness Constraint Entries by Using Command-Line Tools](#)
- [Deleting Attribute Uniqueness Constraint Entries by Using Command-Line Tools](#)

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 7-1](#) on page 7-2.

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. Apply the attribute uniqueness feature by loading 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
orcluniqueobjectclass: 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
orcluniqueobjectclass: 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 10g Release 2 (10.1.2)

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](#)
- [Multimaster Replication Scenario](#)

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.

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](#)
- [Object Classes in the Directory](#)
- [Attributes in the Directory](#)
- [How to Extend the Number of Attributes Associated with Entries](#)
- [Matching Rules in the Directory](#)
- [Syntaxes in the Directory](#)

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](#)

- [Guidelines for Adding, Modifying, and Deleting Object Classes](#)
- [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 recommends that you understand the basic concepts of directory components before attempting to add to or modify the base schema in the directory.

When you add an 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 it 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 a 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:

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

See Also:

- ["About Attribute Management"](#) on page 8-8
- ["Object Classes"](#) on page 2-12 for a conceptual overview of object classes
- ["LDAP Schema Overview"](#) in *Oracle Identity Management User Reference* for a list of schema elements installed with Oracle Internet Directory

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. You can 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`.

Guidelines for Adding, Modifying, and Deleting Object Classes

This section tells you what to keep in mind when adding, modifying, or deleting object classes.

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

Guidelines for Adding Object Classes

When you add object classes, keep the following 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. The Object Identifier must begin with the unique identifier 2.16.840.1.113894 followed by either the Oracle-supplied prefix `.9999` or a site-specific prefix.
- 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 2-12 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 8-1
- ["Managing Object Classes by Using Command-Line Tools"](#) on page 8-7

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.

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 **Oracle Internet Directory Servers**, then ***directory server instance***.

2. Select **Schema Management**. The Schema Management tab pages appear in the right pane.
3. In the right pane, choose **Find Object Classes**. The Find: Object Classes dialog box appears.
4. On the search criteria bar, from the menu farthest to the left, select the property of the object class you want to search for. Options are listed and described in [Table A-22](#) on page A-14.

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 A-23](#) on page A-15.
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 **Oracle Internet Directory Servers**, then *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 **Oracle Internet Directory Servers**, then *directory server instance*.
2. Select **Schema Management**.
3. In the right pane, select the **Object Classes** tab and, in the toolbar, choose **Create**. 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 this one as a template.

4. In the New Object Class dialog box, enter the information in the fields. These are described in [Table A-25](#) on page A-16.
5. Choose **OK**.

See Also:

- ["Object Class Types"](#) on page 2-12
- ["Subclasses, Superclasses, and Inheritance"](#) on page 2-12
- 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 **Oracle Internet Directory Servers**, then ***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 A-25](#) on page A-16.
5. Choose **OK**.

See Also:

- ["Object Class Types"](#) on page 2-12
- ["Subclasses, Superclasses, and Inheritance"](#) on page 2-12

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 8-7 for an example of adding attributes to an auxiliary object class

Deleting Object Classes by Using Oracle Directory Manager

Caution: Oracle 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 Also: The "ldapmodify" command-line tool reference in *Oracle Identity Management User Reference*

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: ( 2.16.840.1.113894.9999.12345 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`
- Gives it an object identifier of `2.16.840.1.113894.9999.12345`.
- Specifies `top` as its superclass
- Specifies `cn` and `sn` as mandatory attributes
- Allows `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:
( 2.16.840.1.113894.9999.12345 NAME 'country' SUP top STRUCTURAL MUST c MAY
  ( searchGuide $ description ) )
-
add: objectclasses
objectclasses:
( 2.16.840.1.113894.9999.12345 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 2-11 for information about attribute options
- ["Managing Entries with Attribute Options by Using Oracle Directory Manager"](#) on page 6-6 and ["Managing Entries with Attribute Options by Using Command-Line Tools"](#) on page 6-9 for instructions on adding and deleting attribute options and for searching for entries containing attribute options
- ["About LDAP Attribute Syntax"](#) in *Oracle Identity Management User Reference* 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 in the base schema, you can add a new attribute or modify an existing one.

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

See Also: ["Attributes"](#) on page 2-8 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: "About LDAP Attribute Syntax" in *Oracle Identity Management User Reference* 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 **Oracle Internet Directory Servers**, then ***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 A-26](#) on page A-16.

See Also: "[Viewing Attributes for a Specific Entry by Using Oracle Directory Manager](#)" on page 6-3 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 **Oracle Internet Directory Servers**, then ***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 A-26](#) on page A-16.
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 A-27](#) on page A-17.
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 A-28](#) on page A-17.
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 8-11.

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

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then *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 A-29](#) on page A-17.
5. Select the **Advanced** tab, and enter values in each of the fields. These are described in [Table A-30](#) on page A-18.
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 8-3.

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 **Oracle Internet Directory Servers**, then ***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 A-29](#) on page A-17. 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 A-30](#) on page A-18.
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 8-3.

Modifying an Attribute by Using Oracle Directory Manager

To modify an attribute by using Oracle Directory Manager:

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then ***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 A-29](#) on page A-17.
6. Select the **Advanced** tab and enter values in each of the fields. These are described in [Table A-30](#) on page A-18.
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 **Oracle Internet Directory Servers**, then ***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 8-14.

You can index only those attributes that have:

- An equality matching rule
 - Matching rules supported by Oracle Internet Directory as listed in "About LDAP Attribute Matching Rules" in *Oracle Identity Management User Reference*
 - Less than 128 characters in their names
-
-

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

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then ***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 8-10.
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 **Oracle Internet Directory Servers**, then ***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:

- The "ldapmodify" command-line tool reference in *Oracle Identity Management User Reference* for a detailed explanation of ldapmodify and its options
- "[Syntaxes in the Directory](#)" on page 8-20 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:

- The "ldapmodify" command-line tool reference in *Oracle Identity Management User Reference* for a detailed explanation of `ldapmodify` and its options
- ["Syntaxes in the Directory"](#) on page 8-20 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 "About LDAP Attribute Matching Rules" in *Oracle Identity Management User Reference*
- No more than 128 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 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: The "ldapmodify" command-line tool reference in *Oracle Identity Management User Reference*

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.sh" command-line tool reference in *Oracle Identity Management User Reference*

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](#)
- [Extending the Number of Attributes for Existing Entries by Creating an Auxiliary Object Class](#)
- [Extending the Number of Attributes for Existing Entries by Creating a Content Rule](#)

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:

- "Oracle Identity Management LDAP Object Classes" in *Oracle Identity Management User Reference* for a list of object classes in the schema installed with Oracle Internet Directory
- [About Object Class Management](#) on page 8-2 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 8-2 for instructions on creating auxiliary object classes
- [Chapter 6, "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.

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 permitted for entries governed by the DIT content rule, in addition to those called for by structural and auxiliary object classes,

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 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 8-18 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 **Oracle Internet Directory Servers**, then *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 A-32](#) on page A-19.
6. Choose **OK**.

Modifying a Content Rule by Using Oracle Directory Manager

To modify a content rule:

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then *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 A-33](#) on page A-19.
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-LABEL OPTIONAL,
auxiliaries        SET (1..MAX) OF OBJECT-CLASS OPTIONAL,
mandatory          SET (1..MAX) OF ATTRIBUTE OPTIONAL,
optional           SET (1..MAX) OF ATTRIBUTE OPTIONAL,
precluded          SET (1..MAX) OF ATTRIBUTE OPTIONAL
}
```

[Table 8-1](#) describes the parameters. Note that the attribute and object class names are case-insensitive.

Table 8–1 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 definition. It must be a unique numeric value that begins with 2.16.840.1.113894 followed by .9999 or a site-specific prefix.
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.
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 **Oracle Internet Directory Servers**, then *directory server instance*.
2. Select **Schema Management**.
3. 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 A–31](#) on page A-18.

Viewing Matching Rules by Using Idapsearch

Use Idapsearch on the subentry `cn=subSchemaSubentry`.

See Also: The "Idapsearch" command-line tool reference in *Oracle Identity Management User Reference*

Syntaxes in the Directory

This section contains these topics:

- [Viewing Syntaxes by Using Oracle Directory Manager](#)
- [Viewing Syntaxes by Using by Using Idapsearch](#)

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 **Oracle Internet Directory Servers**, then ***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 Idapsearch

Use Idapsearch on the subentry `cn=subSchemaSubentry`.

See Also: The "Idapsearch" command-line tool reference in *Oracle Identity Management User Reference*

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](#)
- [Managing Group Entries](#)

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](#)
- [Dynamic Groups](#)
- [Hierarchies](#)
- [Querying Group Entries](#)
- [When to Use Each Kind of Group](#)

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.

For more information on controls used by Oracle Internet Directory, see "About LDAP Controls" in *Oracle Identity Management User Reference*.

Tip: The C API chapter in *Oracle Identity Management Application Developer's Guide*

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/"ou=NewUnit,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.

Do not set `orclConnectByAttribute` or `orclConnectByStartingValue` when using the `labeledURI` attribute method.

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

- 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
<code>orclConnectByAttribute</code>	The attribute that you want to use as the filter for the query—for example, <code>manager</code>
<code>orclConnectByStartingValue</code>	The DN of the attribute you specified in the <code>orclConnectByAttribute</code> attribute—for example, <code>Anne Smith</code>

Do not set `labeledURI` when using the `CONNECT BY` assertion method.

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 Identity Management Application Developer's Guide* for more information about how to develop applications that retrieve values for particular attributes

Limitations of Dynamic Groups in Oracle Internet Directory 10g Release 2 (10.1.2)

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.

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.

For more information on controls used by Oracle Internet Directory, see "About LDAP Controls" in *Oracle Identity Management User Reference*.

See Also: The C API chapter in *Oracle Identity Management Application Developer's Guide*

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

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 Oracle Directory Manager](#)
- [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 9-4.

See Also:

- "[Security Groups](#)" on page 14-3 for instructions on setting access control policies for group entries
- [Globalization Support](#) on page 2-15 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: cn=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 existence of the entry. However, if the attribute value must contain a DN, then the directory server verifies that the DN is specified.

Examples of Dynamic Group Entries

This section provides examples of the two kinds of dynamic group entries.

Example: a Dynamic Group Entry Using the `labeledURI` Attribute

The following is an example of a dynamic group entry using the `labeledURI` attribute.

```
dn: cn=dgroup1
cn: dgroup1
description: this is an example of a dynamic group
labeleduri:ldap://hostname:7777/ou=oid,l=amer,dc=oracle,
dc=dgrptest??sub?objectclass=person
objectclass: orldynamicgroup
objectclass: groupOfUniqueNames
objectclass: top
```

This group will have `uniquemember` values that are the DNs of all entries associated with the object class `person` in the subtree `ou=oid,l=amer,dc=oracle,dc=dgrptest`.

Example: a Dynamic Group Entry Using the `CONNECTBY` Assertion

The following is an example of a dynamic group entry that uses the `CONNECTBY` assertion.

```
dn: cn=dgroup2
cn: dgroup21
description: this is connect by manager assertion dynamic group
orclconnectbyassertionbase: l=amer,dc=oracle,dc=dgrptest
orclconnectbyattribute: mana
orclconnectbystartingvalue: cn=john doe sr.
objectclass: orldynamicgroup
objectclass: groupOfUniqueNames
objectclass: top
```

This dynamic group has unique members with values that are DNs of all the entries whose manager attribute is `cn=john doe sr.` either indirectly or directly. If several

individuals have `cn=john doe JR.` as their manager, and he, in turn, has `cn=john doe SR.` as his manager, then all the lower-level individuals are returned.

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 **Oracle Internet Directory Servers**, then *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 are using the `labeledURI` method for dynamically computing membership in the group, you must set the `labeledURI` attribute, but not the `orclConnectByAttribute` and `orclConnectByStartingValue` attributes. In the **Optional Properties** tab page, in the `labeledURI` field, specify the following:

```
ldap:ldap_URL
```

For example:

```
ldap://my_host/ou=MyNeworganizationalUnit,
o=MyCompany,c=US??sub?(objectclass=person)
```

If you are using the `CONNECT BY` method for dynamically computing membership in the group, you must set the `orclConnectByAttribute` and `orclConnectByStartingValue` attributes, but not the `labeledURI` attribute. In the `orclConnectByAttribute` field, specify the attribute that you want to use as the filter for the query—for example, `manager`. In the `orclConnectByStartingValue` field, specify the DN of the attribute you specified in the `orclConnectByAttribute` attribute—for example, `cn=Anne Smith`.

For information about specifying the other attributes that appear in the **Optional Properties** tab page, see "User and Group Schema Elements" in *Oracle Identity Management User Reference*.

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. If you choose Browse, the Directory: Entry Management dialog box appears. Use this dialog box to search for a particular entry you want to add to the list. Then, in the **Distinguished Name** window of the Directory: Entry Management dialog box, select the entry and 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: orcdynamicgroup
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: orcdynamicGroup
orclConnectByAttribute:attribute_name
orclConnectByStartingValue:DN_of_attribute
```

member: *DN of member 1*

When specifying entries in this syntax, do not use double quotes around distinguished names.

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.

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](#)
- [About Log Messages](#)
- [Setting Debug Logging Levels](#)
- [Setting the Operation Debug Dimension](#)
- [Force Flushing the Trace Information to a Log File](#)

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 * gslsfcStr2Simple *
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: <ul style="list-style-type: none"> ■ 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**.
4. To generate a log for a specific problem, specify the debug logging level on this tab page. Otherwise, you can leave the check boxes on this tab page deselected.

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](#) on page 10-5.

Because debug levels are additive, you need to add the numbers representing the functions that you want to activate, and use the sum of those 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
67108864	Number and identity of clients connected to this server
117440511	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

Table 10–3 (Cont.) Debug Dimension Values for LDAP Operations

Operation Debug Dimension Value	Provides Information Regarding
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. 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, 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

You can, however, view 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.

Example 10–1 Enabling Force Flushing

To enable force flushing by using ldapmodify:

1. Create an LDIF file as follows:

```
dn: cn=dsaconfig,cn=configsets,cn=oracle internet directory
changetype: modify
replace: orcldebugforceflush
orcldebugforceflush: 1
```

2. Load this file by entering 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: "Oracle Identity Management LDAP Attribute Reference" in *Oracle Identity Management User Reference* 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.

This section contains these topics:

- [Structure of Audit Log Entries](#)
- [Position of Audit Log Entries in the DIT](#)
- [Auditable Events](#)
- [Setting the Audit Level](#)
- [Searching for Audit Log Entries](#)
- [Purging the Audit Log](#)

See Also:

- ["Auditable Events"](#) on page 10-9 for a listing of audit levels
- ["Setting the Audit Level"](#) on page 10-10 for instructions on specifying the audit level
- ["Searching for Audit Log Entries by Using Oracle Directory Manager"](#) on page 10-11
- ["Searching for Audit Log Entries by Using ldapsearch"](#) on page 10-12
- The "ldapdelete" command-line tool reference in *Oracle Identity Management User Reference*

Structure of Audit Log Entries

Each audit log entry contains the `orclAuditoc` **object class**. Like all other structural object classes, `orclAuditoc` inherits from `top`. [Table 10-4](#) lists and describes the attributes of the `orclAuditoc` object class.

Table 10-4 Attributes of the `orclAuditoc` Object Class

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

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 Oracle Directory Manager"](#) on page 8-12.

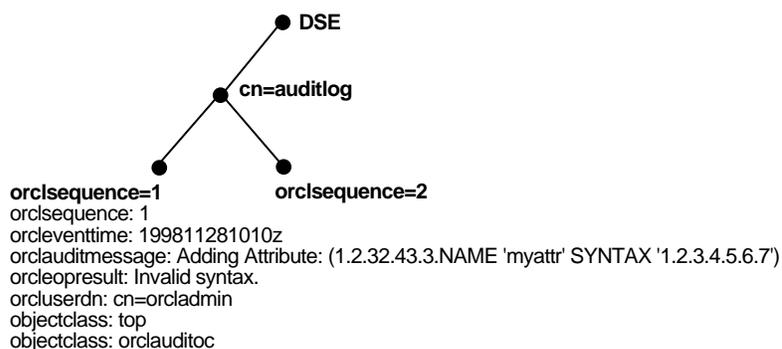
See Also:

- The "catalog.sh" command-line tool reference in *Oracle Identity Management User Reference* for information about cataloged attributes
- "Object Class Types" on page 2-12 for a description of `top`

Position of Audit Log Entries in the DIT

The audit log container is part of the DSE. As shown in [Figure 10–1](#), it holds its entries as children organized according to the `orclsequence` attribute.

Figure 10–1 Sample Audit Log in DSE



Auditable Events

[Table 10–5](#) shows the auditable events and their audit levels. The third column, Audit Levels, contains hexadecimal 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
Access violation	Access denied by access control policy point (ACP)	0x0010
directory-specific entry (DSE) modification	Changes to a DSE (successes or failures)	0x0020
Replication login	Replication server authentication (successes or failures)	0x0040
ACI modification	Changes to an access control list (ACL)	0x0080
User password modification	Modification of user password attribute	0x0100
Add	Idapadd operation (successes or failures)	0x0200

Table 10–5 (Cont.) Auditable Events

Event	Description	Audit Levels
Delete	ldapdelete operation (successes or failures)	0x0400
Modify	ldapmodify operation (successes or failures)	0x0800
ModifyDN	ldapModifyDN operation (successes or failures)	0x1000
bind	Successful user bind cases	0x2000

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 [Table 10–6](#).

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.

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
4. Choose **Apply**.
 5. Restart the directory server instance for the changes to take effect.

See Also: The "oidctl" command-line tool reference in *Oracle Identity Management User Reference* 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 audit masks. For example, suppose you want to audit the events in [Table 10-7](#).

Table 10-7 Example: Setting the Audit Level

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

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: The "oidctl" command-line tool reference in *Oracle Identity Management User Reference* for instructions on how to restart the directory server

Searching for Audit Log Entries

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

Searching for Audit Log Entries by Using Oracle Directory Manager

To use Oracle Directory Manager to view audit log entries:

1. In the navigator pane, expand **Oracle Internet Directory Servers** and **directory server instance**.
2. Select **Audit Log Management**. The corresponding right pane appears.
3. 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.

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 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 A-39](#) on page A-27.
 - 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.
6. To further refine your search, use the buttons in the **Search Criteria** box to enhance the search criteria bar. These are described in [Table A-40](#) on page A-27.
7. Choose **Search**. The results of your search appear in the Distinguished Name box.
8. 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 4-6 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 Also: The "`ldapsearch`" command-line tool reference in *Oracle Identity Management User Reference*

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](#)
- [Oracle Internet Directory Server Manageability Architecture and Components](#)

- [Location of Configuration Information for Oracle Internet Directory Server Manageability](#)
- [Configuring Oracle Internet Directory Server Manageability](#)
- [Configuring Critical Events](#)
- [Using the Oracle Internet Directory Server Manageability Framework Through Oracle Enterprise Manager 10g Application Server Control Console](#)

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: The chapter on Oracle Directory Integration and Provisioning concepts and components in *Oracle Identity Management Integration Guide*

You can view monitored information by using the Oracle Enterprise Manager 10g Application Server Control Console.

Note:

The Application Server Control Console does not display the port status information for Oracle directory servers running only in SSL mode.

See Also:

- Online help for Oracle Enterprise Manager 10g Application Server Control Console
- The chapter about administration tools in the *Oracle Application Server Administrator's Guide*

Oracle Internet Directory Server Manageability Architecture and Components

The relationship between the various components of directory server manageability is explained in [Figure 10-2](#) and the accompanying text in [Table 10-8](#).

Figure 10-2 Architecture of Oracle Internet Directory Server Manageability

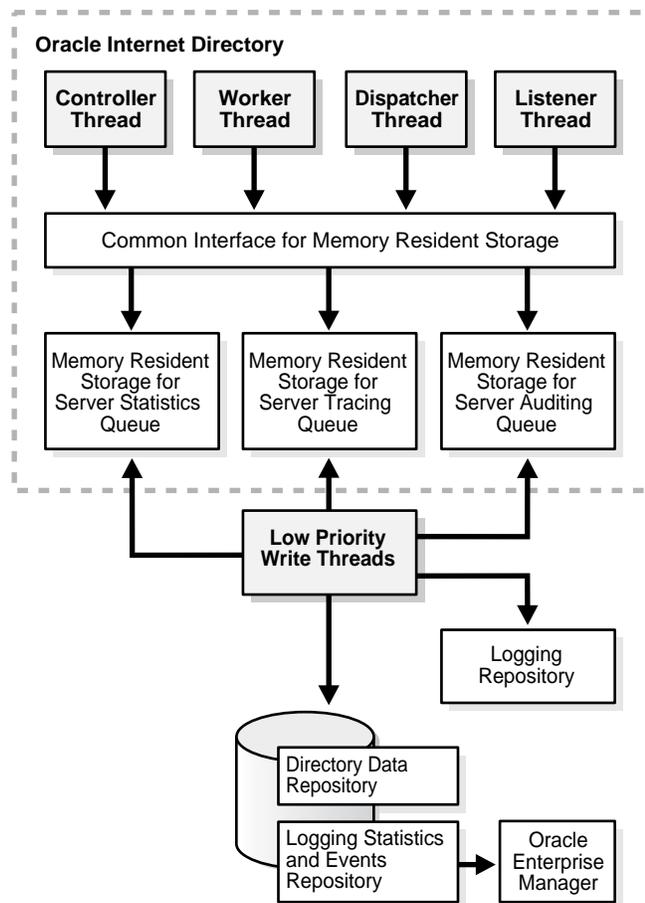


Table 10–8 Components of Oracle Internet Directory Server Manageability

Component	Description
Oracle Internet Directory	<p>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.</p> <p>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.</p> <p>See Also: "An Oracle Directory Server Instance" on page 2-3 for a description of the directory server</p>
Memory Resident Storage	This is a local process memory. The Oracle Internet Directory Server 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 10g Application Server Control Console	The Application Server Control Console 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	<p>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:</p> <ul style="list-style-type: none"> ■ Normal LDAP operations to store and retrieve the information ■ Existing access control policies to manage the security of the gathered information <p>The directory manageability framework isolates the gathered information from the directory data by storing the two separately.</p>

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 statistics, 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: "Oracle Identity Management LDAP Attribute Reference," in *Oracle Identity Management User Reference* 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: Online help for Oracle Enterprise Manager 10g Application Server Control Console 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 10-9](#).

Table 10-9 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 10–9 (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 10g Application Server Control Console

To exploit the features of Oracle Internet Directory Server Manageability, you use Oracle Enterprise Manager 10g Application Server Control Console as explained in this section.

See Also: For information about stopping and starting Oracle Enterprise Manager 10g Application Server Control Console, see *Oracle Application Server Administrator's Guide*.

Enabling Information Collection by Using Oracle Enterprise Manager 10g Application Server Control Console

To enable information collection by using Oracle Enterprise Manager 10g Application Server Control Console:

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 10g Application Server Control Console

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 the fields in [Table 10–10](#).

Table 10–10 Fields in the Start a New LDAP Server Instance Window of the Application Server Control Console

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

Table 10–10 (Cont.) Fields in the Start a New LDAP Server Instance Window of the Application Server Control Console

Column	Description
Port Number	The port number you assign to the directory server instance if the default port number is not used

- 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.
- Choose **Start**.

Stopping a Directory Server Instance by Using Oracle Enterprise Manager 10g Application Server Control Console

To stop a directory server instance:

- In the Oracle Internet Directory main window, in the **LDAP Instances** section, select the directory server instance you want to stop.
- Choose **Stop**.

Restarting a Directory Server Instance by Using Oracle Enterprise Manager 10g Application Server Control Console

To restart a directory server instance:

- In the Oracle Internet Directory main window, in the **LDAP Instances** section, select the server you want to restart.
- Choose **Restart**. The Restart an LDAP Server Instance window displays the fields listed in [Table 10–11](#).

Table 10–11 Fields in the Restart an LDAP Server Instance Window of the Application Server Control Console

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

- Select a configuration. If the default port is not available, then, in the **Port Number** column, enter a port number.
- Choose **Start**.

Viewing Directory Server Activities by Using Oracle Enterprise Manager 10g Application Server Control Console

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 10g Application Server Control Console

You can view directory server operations over the course of the specified time period of statistics collection by using Application Server Control Console. 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.

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](#)
- [Backing Up and Restoring a Large Directory](#)

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:

```
bulkload.sh -connect connect_string -check -generate -load -restore \  
-append /complete_path/backup.ldif
```

Backing Up and Restoring a Large Directory

For instructions on backing up and restoring a large directory, see *Oracle Application Server Administrator's Guide*.

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](#)
- [Data Privacy and Oracle Internet Directory](#)
- [Authorization in Oracle Internet Directory](#)
- [Authentication in Oracle Internet Directory](#)
- [Protection of User Passwords for Directory Authentication](#)
- [Password Policies in Oracle Internet Directory](#)
- [Authentication by Using Simple Authentication and Security Layer \(SASL\)](#)

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 direct authentication available within Oracle Internet Directory, and about how SASL-enabled clients authenticate to a directory server. The three kinds of direct authentication options are:

- 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:

- Digest MD5—The required authentication mechanism within LDAP Version 3 (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—Mechanism using SSL mutual authentication. 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. Certificate authentication can take the following forms:
 - * Exact match—the subject DN in the client certificate is compared with the user DN in the directory. If the two values match, a bind occurs.
 - * Certificate hash—The client certificate is hashed and is then compared with the hashed value of the certificate stored in the directory. If the two values match and only one DN is associated with the pair, a bind occurs. If two or more DNs are associated, an error will be returned because certificate hash and user DN is an n-to-1 mapping and not a 1-to-n mapping. That is, you can have many certificates associated with one DN, but only one DN associated with a certificate.
 - * Exact match/certificate hash—An exact-match search is performed first. If this search yields nothing, a certificate hash is performed.

To choose one of these methods, edit the DSA configuration parameter `orclpkimatchingrule` as prescribed in "Oracle Identity Management LDAP Attribute Reference" in *Oracle Identity Management User Reference*. (For authentication, an `orclpkimatchingrule` value of 3 or 4 is equivalent to a value of 2.)

Notes:

- The introduction in 10g Release 2 (10.1.2) of a certificate hash value requires that user certificates be upgraded from earlier releases. To learn how to upgrade certificates, see the "upgradecert.pl" command-line tool reference in *Oracle Identity Management User Reference*.
- You can search for the binary attribute `usercertificate`. To learn how to conduct a search, see [Appendix G, "Searching the Directory for User Certificates"](#).

See Also:

- [Authentication by Using Simple Authentication and Security Layer \(SASL\)](#) on page 12-7
- 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

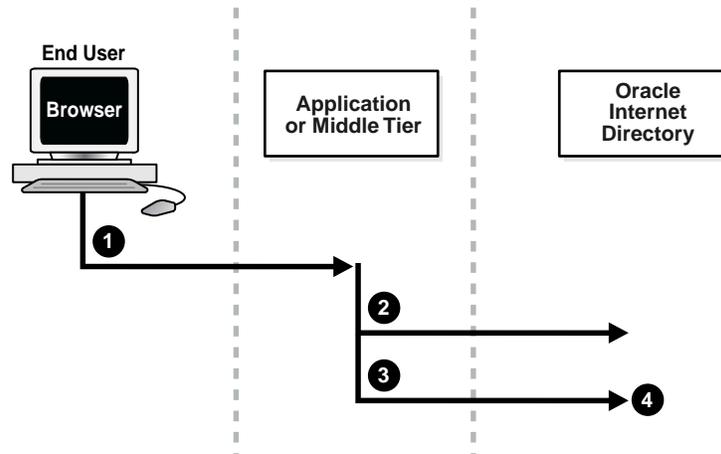
Indirect Authentication

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 14-8

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.

See Also: [Chapter 28, "Setting Up the Customized External Authentication Plug-in"](#)

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: [Chapter 16, "Directory Storage of Password Verifiers"](#)

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 12-3 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 a digest-challenge that includes various Digest-MD5 authentication options that it supports and a special token.
2. The client selects an authentication option, then sends a digest-response to the server indicating the option it has selected. The response includes some secure tokens and a client credential in encrypted format. This allows it to authenticate itself to the server.
3. The directory server then decrypts and verifies the client credential from the 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. The authorization identity is derived as follows:

- In case of exact match, the authorization identity is derived from the client authentication credentials in the system providing external authentication—for example, the client SSL certificate.

- In case of exact match, the authorization identity is derived from the client authentication credentials in the system providing external authentication—for example, the client SSL certificate

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.

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](#)
- [Limitations of the Use of SSL in 10g Release 2 \(10.1.2\)](#)
- [Configuring and Testing Oracle Internet Directory With SSL](#)
- [Other Components and SSL](#)

See Also: ["Security"](#) on page 2-14 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.

[Table 13-1](#) lists the SSL cipher suites supported by Oracle Internet Directory and their corresponding authentication, encryption, and data integrity mechanisms.

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

Table 13–1 (Cont.) SSL Cipher Suites Supported in Oracle Internet Directory

Cipher Suite	Authentication	Encryption	Data Integrity
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.

Limitations of the Use of SSL in 10g Release 2 (10.1.2)

In Oracle Internet Directory 10g Release 2 (10.1.2), the Oracle directory replication server cannot communicate directly with an SSL-enabled LDAP server that supports two way (mutual) authentication. The replication server startup will fail and hang if the LDAP server is configured for SSL mutual authentication.

See Also: [Chapter 5, "Oracle Directory Server Administration"](#) for instructions on how to configure server instances

Configuring and Testing Oracle Internet Directory With SSL

Use Oracle Wallet Manager to configure Oracle Internet Directory for SSL. To test the connection, use either the command line or Oracle Directory Manager.

Note: By default, the SSL port that is defined in configuration set 0 is set to authentication mode 1 (encryption only). Do not configure the SSL port of configset 0 with a authentication mode other than 1. Doing so will break Oracle Delegated Administration Services and other applications that expect to communicate with Oracle Internet Directory on the encrypted SSL port.

This section contains these topics:

- [Configuring SSL Parameters](#)
- [Configure Oracle Internet Directory for SSL](#)
- [Testing SSL Connections From the Command Line](#)
- [Testing SSL Connections With Oracle Directory Manager](#)

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 3031. To allow the same instance to run non-secure connections concurrently, set SSL Enable to 2: the default non-secure port is 3060.

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.

This section contains these topics:

- [Configuring SSL Parameters by Using Oracle Directory Manager](#)
- [Configuring SSL Parameters by Using Command-Line Tools](#)

See Also:

- ["Managing Server Configuration Set Entries"](#) on page 5-1 for instructions on how to set these parameters
- "OID Configuration Schema Elements," in *Oracle Identity Management User Reference* 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.

Viewing and Modifying SSL Configuration Parameters To view and modify SSL configuration parameters:

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then **directory server instance**, then **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 A-41](#) on page A-28.

See Also: ["Managing Server Configuration Set Entries by Using Oracle Directory Manager"](#) on page 5-3 for information about changing parameters in a configuration set entry

Configuring SSL Parameters by Using Command-Line Tools

For information about configuring SSL parameters from the command line, see:

- "[Managing Server Configuration Set Entries by Using Command-Line Tools](#)" on page 5-5
- "Oracle Internet Directory Server Administration Tools" in *Oracle Identity Management User Reference* for instructions on using the `-p`, `-U`, and `-w` flags to `ldapadd` and related commands to configure SSL

Configure Oracle Internet Directory for SSL

Configure the server side LDAP server for SSL as follows:

1. Start the Oracle Wallet Manager

On Unix, set the `DISPLAY` environment variable and type:

```
owtm
```

On Windows, start the program by using either:

- **Start**, then **Programs**, then **ORACLE_HOME**, then **Network Administration**, then **Wallet Manager**
 - **Start**, then **Programs**, then **ORACLE_HOME**, then **Integrated Management Tools**, then **Wallet Manager**
2. Select **Wallet** from the top menu bar and then **New**.
Choose and confirm the password.
 3. A new empty wallet has been created.
Select **YES** to create a certificate request.
 4. Enter the required information.

See Also: *Oracle Advanced Security Administrator's Guide* for information on using Oracle Wallet Manager.

5. Choose **OK**.

An Oracle Wallet Manager dialog box informs you that a certificate request was successfully created. You can either copy the certificate request text from the body of this dialog panel and paste it into an e-mail message to send to a certificate authority, or you can export the certificate request to a file.

6. Choose **Operations**, then **Export Certificate Request** from the menu bar.
The Export Certificate Request dialog box appears.
7. Enter a file name for the request, such as `usercert.req`.
8. Save the wallet.

Note: When saving the wallet on Windows 2000, choose a directory path that does not contain spaces. Do not store the wallet in the default location, *Documents and Settings\oracle\wallets*.

9. Send the newly-created certificate request to your certificate authority.

See Also:

- *Oracle Application Server Certificate Authority Administrator's Guide*
- MetaLink Note: 178806.1: *How to get SSL certificates from a Microsoft Certification Services CA*, on Oracle MetaLink, <http://metalink.oracle.com>.

for information on certificates from a Microsoft Certification Services Certificate Authority.

10. You should receive a user certificate and, if needed, a trusted certificate from your certificate authority. If your CA is not in Oracle Wallet Manager's default list, you must import a trusted certificate from your CA before you can import the user certificate.
 - a. To import the trusted certificate, choose **Operations**, then **Import Trusted Certificate** from the menu bar. The **Import Trusted Certificate** dialog panel appears. Choose either to paste the certificate in base64 format or to select a file containing the trusted certificate. Your new CA will appear in the list of Trusted Certificates.
 - b. To import the user certificate, choose **Operations**, then **Import Trusted Certificate** from the menu bar. The **Import Certificate** dialog box appears. Choose either to paste the certificate in base64 format or to select a file containing the trusted certificate.
11. Select **Wallet** and save the wallet by selecting **Wallet**, then **Save**. Enable Auto Login by choosing **Wallet** from the menu bar, then choosing the check box next to the **Auto Login** menu item. A message at the bottom of the window displays **Auto Login Enabled**. A file called *cwallet.sso* is now present in your wallet directory.

Note: As of Oracle Internet Directory Release 9.0.2, only wallets in encrypted format, such as *cwallet.sso*, are supported. For that reason, you must use Oracle Wallet Manager to open the wallet and to enable Auto Login before you start an SSL instance.

12. Open the Oracle Directory Manager and choose to add a new **Configuration Set**. Do not modify the **Default Configuration Set**.

Select the **SSL Setup** tab and enter the location of the wallet. For UNIX, the URL format is:

```
file://path/directory_of_wallet
```

For example:

```
file://etc/ORACLE/WALLET
```

For Windows, the URL format is:

```
file:\device:\path\wallet_directory
```

For example:

```
file:d:\wallet
```

Choose the SSL authentication method and configure the SSL port. The authentication methods are:

SSL Authentication Method	Authentication Behavior
No SSL Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. Only SSL encryption and decryption is used.
SSL Server Authentication	The directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.
SSL Client and Server Authentication	The client and server authenticate themselves to each other and send certificates to each other.

Choose the port for the SSL instance for the release.

See Also: ["Process Control of Oracle Internet Directory Components"](#) on page 4-8.

13. You now have three configuration sets: `DefaultConfigset` with a default SSL port and a default non-SSL port, `default Configset1` and your `newConfigset2` with a unique SSL port and unique non-SSL port.

On Windows systems, you must perform an extra configuration step. You must change the login account of the Oracle Directory Service from a local system account to the account of the user who owns the wallet. This user must be member of `Administrator Group`. Change the account as follows:

- a. On Windows, choose **Start**, then **Settings**, then **Control Panel**, then **Administrative Tools**, then **Services**.
 - b. Click **PROPERTIES/LOGON**.
 - c. Change from **Local System Account** to the account you logged in as when you created the Wallet. Stop and restart the service.
14. Start the Oracle Internet Directory instances so that Oracle Delegated Administration Services and other applications requiring SSL in encrypted mode can operate normally.
- a. Open a browser to the Oracle Enterprise Manager Web site and drill down into the Oracle Internet Directory processes. This page shows the running processes.
 - b. Click the Button **Start New Instance**. The new configuration set will be listed.
 - c. Select the **Set Number** to be started and click **Start**.
 - d. After the instance is started, click **OK** and the Oracle Internet Directory instances page will be displayed. The new instance will be shown in the list as started.

From this point on, the standard commands

```
opmnctl startall
opmnctl stopall
```

will automatically manage the Oracle Internet Directory instances.

15. You now have Oracle Internet Directory running and listening on four ports.

On a UNIX system, you can run the `$ORACLE_HOME/ldap/bin/ldapcheck` command to view the additional `oidldapd` dispatcher and server processes. The

debugging logs for the SSL instance are `oidldapd02.log` and `oidldapd02sXXXXXX.log`, respectively.

Testing SSL Connections From the Command Line

You can use the `ldapbind` command to test SSL connections. On UNIX, the syntax is:

```
ldapbind -D cn=orcladmin -w welcome -U authentication_mode -h host -p SSL_port \  
-W "file://DIRECTORY_CONTAINING_WALLET" -P wallet_password
```

and on Windows, the syntax is:

```
ldapbind -D cn=orcladmin -w welcome -U authentication_mode -h host -p SSL_port \  
-W "file:device:\DIRECTORY_CONTAINING_WALLET" -P wallet_password
```

where *authentication_mode* is one of:

Number	Authentication
1	No SSL authentication required.
2	One-way (server only) SSL authentication required.
3	Two-way (client and server) SSL authentication required.

See Also: The "ldapbind" command-line tool reference in *Oracle Identity Management User Reference*.

Testing SSL With Encryption Only

Use this method to test an SSL configuration with no SSL authentication required. The syntax is:

```
ldapbind -D cn=orcladmin -w password -U 1 -h host -p SSL_Port
```

Testing SSL With Server Authentication

Use this method to test an SSL configuration with SSL server authentication configured. A client can request either server authentication or no authentication.

For an anonymous bind with server authentication, the syntax is:

```
ldapbind -U 2 -h host -p port -W "file:DIRECTORY_CONTAINING_WALLET" \  
-P wallet_password
```

For a bind with user "cn=orcladmin" and server authentication, the syntax is:

```
ldapbind -D cn=orcladmin -w password -U 2 -h host -p port \  
-W "file:DIRECTORY_CONTAINING_WALLET" -P wallet_password
```

For a bind without SSL authentication, the syntax is:

```
ldapbind -D cn=orcladmin -w password -U 1 -h host -p SSL_Port
```

Testing SSL With Client and Server Authentication

Use this method to test an SSL configuration with SSL client and server authentication configured.

As of Oracle Internet Directory 10g Release 2 (10.1.2), Oracle Internet Directory supports the Certificate Matching Rule. The DN and password passed on the

ldapbind command line are ignored. Only the DN from the certificate or the certificate hash is used for authorization.

See Also: ["Direct Authentication"](#) on page 12-3.

To bind with user "cn=orcladmin", the syntax is:

```
ldapbind -D cn=orcladmin -w password> -U 3 -p port \  
-W "file:DIRECTORY_CONTAINING_WALLET" -P wallet_password
```

or

```
ldapbind -D cn=orcladmin -w password -U 2 -h host -p port \  
-W "file:DIRECTORY_CONTAINING_WALLET" -P wallet_password
```

To use the bind DN (Distinguished Name) from the client certificate, the syntax is:

```
ldapbind -U 3 -h host -p port -W "file:DIRECTORY_CONTAINING_WALLET" \  
-P wallet_password
```

or

```
ldapbind -U 2 -h host -p port -W "file:DIRECTORY_CONTAINING_WALLET" \  
-P wallet_password
```

Testing SSL Connections With Oracle Directory Manager

To test the SSL connection with the Oracle Directory Manager, perform the following steps:

1. Start Oracle Directory Manager.
At the login screen, click the Network Icon and add the new SSL instance.
Choose the hostname and the port number of your configured SSL instance.
2. It should show **AVAILABLE**. Highlight it and click **SELECT**.
3. Click the **SSL** tab and fill in the wallet location of the user and password. For Windows, specify the SSL location as
`file:device:\wallet_directory_path`
For UNIX, specify the SSL Location as
`file://wallet_directory_path`
For **SSL Password**, specify your wallet password.
For **SSL Authentication Level**, specify your configured authentication level.
4. Click the **Credentials** tab. Make sure the SSL check box is checked. If you omit this step, Oracle Directory Manager might hang.
5. Specify values for **User** and **Password**.

Other Components and SSL

At installation, Oracle Internet Directory starts up with configset0, which specifies dual mode. That is, some components can access Oracle Internet Directory using non-SSL connections, while others use SSL when connecting to the directory. By default, Oracle Application Server components are configured to run in this dual

mode environment when communicating with Oracle Internet Directory. If you wish, you can remove the non-SSL mode and change all middle-tier instances to use SSL. For more information, please refer to the section on changing Oracle Internet Directory from dual mode to SSL mode in *Oracle Application Server Administrator's Guide*.

Enterprise User Security or a customer application might need an SSL channel with a different configuration from configset0. For example, it might need SSL server authentication mode or SSL mutual authentication mode. In this case, you must configure an additional SSL mode port another configuration set so that an additional Oracle Internet Directory LDAP instance listens at that port.

Note: You should never modify the SSL mode of configset0. The modification might conflict with the default configuration of some Oracle Application Server components. You should use a different configuration set for a new SSL setup.

For more information about Enterprise User Security SSL configuration, please see the section on enterprise user security configuration in *Oracle Database Enterprise User Administrator's Guide*.

Examples:

1. A configuration set for SSL server authentication mode:

```
cn=configset2, cn=osldapd, cn=subconfigsubentry
cn=configset2
objectclass=top
objectclass=orclConfigSet
objectclass=orclLDAPSubConfig
orclsslauthentication=32
orclsslenable=2
orclsslwalleturl=file:/ade/qdinh_newld/oracle/work/ldap/lrgsrg
orclsslport=6060
orclnonsslport=8019
orclserverprocs=1
```

2. A configuration set for SSL mutual authentication mode:

```
cn=configset3, cn=osldapd, cn=subconfigsubentry
cn=configset3
objectclass=top
objectclass=orclConfigSet
objectclass=orclLDAPSubConfig
orclsslauthentication=64
orclsslenable=2
orclsslwalleturl=file:/ade/qdinh_newld/oracle/work/ldap/lrgsrg
orclsslport=7001
orclnonsslport=8029
orclserverprocs=1
```

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`.

Note: As of Oracle Internet Directory 10g Release 2 (10.1.2), the super user is subject to access control policies like any other user. The new ACL syntax changes for super user restriction cannot be administered through Oracle Directory Manager.

The access control policies described in this chapter allow the delegation of coarse-grained privileges across an entire realm. If you need finer-grained delegation, you might wish to use Oracle COREid Access and Identity. Please see Chapter 2 of *Oracle COREid Access and Identity Administration Guide*.

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 2-14 and [Chapter 12, "Directory Security Concepts"](#) for a conceptual explanation before you begin implementing and administering access control policies
- [Appendix C, "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 14-10

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 the ACL directives within the entry for that entity. You do this by using a user-modifiable operational attribute called `orclEntryLevelACI`. This attribute contains ACL directives only for 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 object 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 14-6 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. If, however, the `orclACI` or `orclEntryLevelACI` attribute of a subordinate ACP contains the keyword `DenyGroupOverride`, the higher level ACP does not override the subordinate ACP. Use `DenyGroupOverride` to restrict super user access through privileged groups.

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.

Add the `DenyGroupOverride` keyword to an ACI to deny access to members of privileged groups.

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 a security group object class—either `orclACPgroup` or `orclPrivilegeGroup`. Even if a group is a member of a security group, the directory server does not consider it for access control purposes unless it is associated with a 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 sample group of entries in [Table 14-1](#), each of which, with the exception of Group 4, is marked as a privilege group (`objectclass:orclprivilegegroup`). You can set access control policies that apply to the members of `group1`, `group2`, and `group3`.

Table 14–1 Sample Security Groups

Group	Entry
Group 1	dn: cn=group1,c=us cn: group1 objectclass: top objectclass: groupofUniqueNames objectclass: orclPrivilegeGroup uniquemember: cn=mary 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 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 Group 4 because Group 4 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 Group 1 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 6-5 or "[Example: Adding a User Entry by Using Oracle Directory Manager](#)" on page 6-4 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)
```

The *object* part of an access control directive may also include special keywords. These are:

- `DenyGroupOverride`, which prevents access from being overridden by higher level ACPs
- `AppendToAll`, which causes the subject of an ACI to be added to all other ACIs in that ACP during evaluation.

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 C, "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 being granted access
- The bind mode—that is, the authentication mode used to verify the identity of that entity
- The added object constraint, which limits the 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
- The keyword `SuperUser`, which matches the `SuperUser` DN specified in the directory.
- 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:

1. Looks up the value for her `manager` attribute and finds it to be John Doe
2. Verifies that the bind DN matches the `manager` attribute
3. 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 C, "The Access Control Directive Format"](#)
and [Appendix B, "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
- Read/noread
- Selfwrite/noselfwrite
- Write/nowrite
- Add/noadd
- Proxy/noproxy
- Browse/nobrowse
- Delete/nodelete

Note that each access level can be independently granted or denied. The `noxxx` means `xxx` permission is denied.

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

Table 14–2 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
Selfwrite	Right to add yourself to, delete yourself from, or modify your 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: <code>access to attr=(member) by dnattr=(member) (selfwrite)</code> The <code>dnattr</code> selector indicates that the access applies to entities listed in the member attribute. The <code>selfwrite</code> 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 <code>ldapsearch</code> 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

Table 14-3 lists the various LDAP operations and the access required to perform each one.

Table 14-3 *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	<ul style="list-style-type: none"> ▪ 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 states described in Table 14-4:

Table 14-4 *Attribute States During ACL Evaluation*

State	Description
Resolved with permission	The required access for the attribute has been granted in the ACI.
Resolved with denial	The required access for the attribute has been explicitly denied in the ACI.

Table 14–4 (Cont.) Attribute States During ACL Evaluation

State	Description
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

Oracle Internet Directory, enables you to define more than one ACI in the ACP of an object. It 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 this 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 = ( LDAPFILTER ) ]
```

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 1: Reset the Default Security Configuration](#)" on page 3-1

Note: As of Oracle Internet Directory 10g Release 2 (10.1.2), the super user is subject to access control policies like any other user. The new ACL syntax changes for super user restriction cannot be administered through Oracle Directory Manager.

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: "Oracle Identity Management Command-line Tool Reference" in *Oracle Identity Management User Reference* 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 14-14, then you can locate and view an ACP as follows:

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then **directory server instance**, then **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 A-3](#) on page A-4.

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 14-14, then you can locate and view an ACP as follows:

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then **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 14-14, then begin as follows:
 - a. In the navigator pane, expand **Oracle Internet Directory Servers**, then **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 14-14, then begin as follows:

- a. In the navigator pane, expand **Oracle Internet Directory Servers**, then **directory server instance**, then **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 A-39](#) on page A-27.
 - 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 A-39](#) on page A-27.
 - 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 A-4](#) on page A-4.
- 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 A-5](#) on page A-4.
- b. Specify the entity or entities to whom you are granting access. The options are described in [Table A-6](#) on page A-4.
5. Select the **Access Rights** tab page.
- Specify the kinds of rights to be 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
6. 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 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 A-39](#) on page A-27.
 - 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 A-4](#) on page A-4.

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 A-5](#) on page A-4.
 - b. Specify the entity or entities to whom you are granting access. The options are described in [Table A-6](#) on page A-4.
 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 A-7](#) on page A-5.
 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 by Using the ACP Creation Wizard](#)
- [Task 3: Configure Content Access Items by Using the ACP Creation Wizard](#)

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 14-14, then begin as follows:
 - a. In the navigator pane, expand **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 14-14, then begin as follows:

- a. In the navigator pane, expand **Oracle Internet Directory Servers**, then **directory server instance**, then **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 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 A-39](#) on page A-27.
 - 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 ACIs 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 A-39](#) on page A-27.
 - 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 A-4](#) on page A-4.
 - b. To specify the type of encryption: From the **Encryption Choice** list, choose an encryption method. The options are described in [Table A-5](#) on page A-4.
 - c. Specify the entity or entities to whom you are granting access. Options are described in [Table A-6](#) on page A-4.
 - d. Click **Next**. A Structural Access Item dialog box prompts you for access rights information.
5. Specify the kinds of rights that 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 A-27](#) on page A-17.
 - 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 A-4](#) on page A-4.
 - b. To specify the type of encryption, from the **Encryption Choice** list, choose an encryption method. The options are described in [Table A-5](#) on page A-4.
 - c. Specify the entity or entities to whom you are granting access. Options are described in [Table A-6](#) on page A-4.

- 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 the kinds of rights to be granted. These are described in [Table A-7](#) on page A-5.
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

If you configured Oracle Directory Manager always to display ACPs, as described in "[Configuring the Display of ACPs in Oracle Directory Manager](#)" on page 14-14, then begin as follows:

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then **directory server instance**, then **Access Control Management**.
2. 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.
3. In the navigator pane, 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 14-14, then begin as follows:

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then **directory server instance**, then **Access Control Management**.
2. 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 14-16 for instructions about adding structural access items

To modify structural access items:

1. In the **Structural Access Items** window, select the item you want to modify, and, just below the **Structural Access Items** window, click **Edit**. The Structural Access Item dialog box appears.
2. Use the **Entry Filters** tab page to narrow the set of entries to which you are granting access. If you want all entries below the ACP to be governed by the ACP, proceed to the next step.

You might choose an entry based on one or more attributes. For example, you might choose to search for all those whose title is secretary, or for all those whose title is manager and whose organization unit is Americas.

In the **Criteria** window of the **Entry Filters** tab page, use the search criteria bar to select an attribute, enter a value for that attribute, and specify a filter for matching the specified attribute with the value you entered. To do this:

- a. From the menu at the left end of the 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 A-27](#) on page A-17.
 - 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 A-39](#) on page A-27.
 - 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 A-4](#) on page A-4.
 - b. To specify the type of encryption: From the **Encryption Choice** list, choose an encryption method. The options are described in [Table A-5](#) on page A-4.
 - c. Specify the entity or entities to whom you are granting access. The options are described in [Table A-6](#) on page A-4.
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 14-17 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 A-39](#) on page A-27.
 - 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 A-4](#) on page A-4.
 - b. To specify the type of encryption, from the **Encryption Choice** list, choose an encryption method. The options are described in [Table A-5](#) on page A-4.
 - c. Specify the entity or entities to whom you are granting access. The options are described in [Table A-6](#) on page A-4.
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 A-7](#) on page A-5.
6. Click **OK**.

Granting Entry-Level Access by Using Oracle Directory Manager

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

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then *directory server instance*, then **Entry Management**.
2. In the navigator pane, select the entry to display its properties in the right pane
3. 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 14-21.
4. 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 **Oracle Internet Directory Servers**, then *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



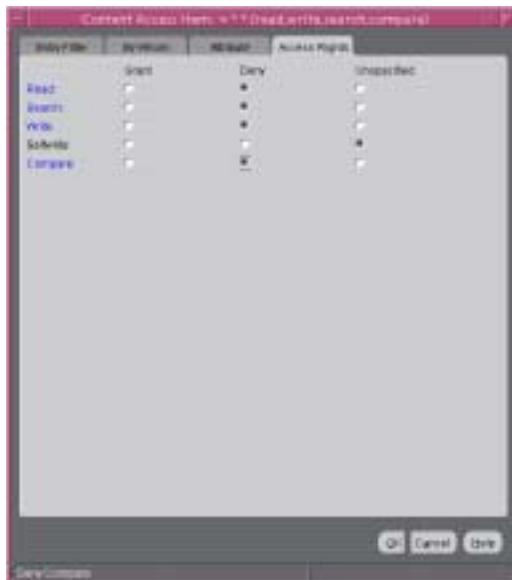
3. Select the **Attribute** tab page. This page has two fields. The first has two choices: **EQ** (equals) and **NEQ** (not equals). The second sets the attribute.
Select **EQ** and select **userPassword**. The **Attribute** tab page should look like [Figure 14-5](#).

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 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 **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 Content Access Item: 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 14-1, directory access control policy information is represented as user-modifiable operational attributes. You can manage it by using command-line tools, including `ldapmodify` and `ldapmodifymt`, to set and alter the values of these attributes.

To directly edit the ACI, you should understand the format and semantics of the directory representation of the ACI as described in [Appendix C, "The Access Control Directive Format"](#).

This section contains these topics:

- [Example: Restricting the Kind of Entry a User Can Add](#)
- [Example: Setting Up an Inheritable ACP by Using ldapmodify](#)
- [Example: Setting Up Entry-Level ACIs by Using ldapmodify](#)
- [Example: Using Wild Cards](#)
- [Example: Selecting Entries by DN](#)
- [Example: Using Attribute and Subject Selectors](#)
- [Example: Granting Read-Only Access](#)
- [Example: Granting Selfwrite Access to Group Entries](#)
- [Example: Defining a Completely Autonomous Policy to Inhibit Overriding Policies](#)

See Also:

- "LDIF File Formatting Rules and Examples" in *Oracle Identity Management User Reference* for information about how to format input by using **LDIF**, the required input format for line mode commands
- The "ldapmodify" command-line tool reference in *Oracle Identity Management User Reference* for information about how to run ldapmodify
- [Appendix C, "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.

In the ACP at `dc=com`, the `orclACI` attribute is specified as follows:

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

Note that, in order to enable reading the attributes, you must grant permission to browse the 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.

The `orclACI` attribute of `dc=acme,dc=com` is specified as follows:

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

The `orclACI` attribute of `dc=us, dc=acme,dc=com` is specified as follows:

```
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 your manager and viewed by yourself. No one else has access to the salary attribute.
- The `userPassword` attribute can be viewed and modified by yourself and the administrator. Others can only compare this attribute.
- The `homePhone` attribute can be read and written by yourself 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.

The `orclACI` attribute of `dc=us,dc=acme,dc=com` is specified as follows:

```
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.

The `orclACI` attribute of `dc=acme, dc=com` is specified as follows:

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

The `orclACI` attribute of `dc=us, dc=acme, dc=com` is specified as follows:

```
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 enables 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.

The `orclEntryLevelACI` attribute of the group entry is specified as follows:

```
access to attr=(member)
by dn=".*, dc=us,dc=acme,dc=com" (selfwrite)
```

Example: Defining a Completely Autonomous Policy to Inhibit Overriding Policies

This example denies group override. It uses the following DNs:

Table 14–5 DNs Used in Example

Container	DN
Naming context to be restricted from Group overriding policies	<code>c=us</code>
User container	<code>cn=users, c=us</code>
Sensitive data	<code>cn=appdata</code>
User admin group for this naming context	<code>cn= user admin group, cn=users, c=us</code>
Security admin group or this naming context	<code>cn= security admin group, cn=users, c=us</code>
Global password admin group for all naming contexts that reset passwords	<code>cn=password admin group</code>

The policy requirements for `c=us` are as follows:

- Users can browse and read their information.
- The user security admin can modify the information under `c=us` except for passwords and ACPs.

- The security admin group can modify policies under `c=us`.
- The global password admin and the user can reset a password.
- All other users have no permissions.
- This policy cannot be overridden.

Required ACP:

```
Access to entry DenyGroupOverride
by dn=".*,c=us" (browse,noadd,nodelete)
by group="cn=User admin group,cn=users,c=us" (browse,add,delete)
```

```
Access to attr=(orclaci) DenyGroupOverride
by group="cn=security admin group,cn=users,c=us" (search,read,write,compare)
by * (none)
```

```
Access to attr=(userpassword) DenyGroupOverride
by self (search,read,write,compare)
by group="cn=password admin group" (search,read,write,compare)
by * (none)
```

```
Access to attr=(*) DenyGroupOverride
by self (search,read,nowrite,compare)
by group="cn= User admin group,cn=users,c=us" (search,read,write,compare)
by * (none)
```


Part III

Directory Security

This part explains how to:

- Secure data within the directory
- Establish access controls for administering applications in enterprises and hosted environments
- Establish and manage policies governing passwords
- Manage password verifiers used to authenticate users to other Oracle components
- Store data for users, groups, and services in one repository, and delegate the administration of that data to various administrators

It contains these chapters:

- [Chapter 12, "Directory Security Concepts"](#)
- [Chapter 13, "Secure Sockets Layer \(SSL\) and the Directory"](#)
- [Chapter 14, "Directory Access Control"](#)
- [Chapter 15, "Password Policies in Oracle Internet Directory"](#)
- [Chapter 16, "Directory Storage of Password Verifiers"](#)
- [Chapter 17, "Delegation of Privileges for an Oracle Technology Deployment"](#)

Password Policies in Oracle Internet Directory

Password policies are sets of rules that govern how passwords are used. This chapter contains these topics:

- [About Password Policies](#)
- [Managing Password Policies](#)
- [Password Policy Error Messages](#)

About Password Policies

This section contains these topics:

- [What a Password Policy Is](#)
- [Default Password Policy](#)
- [Directory Server Verification of Password Policy Information](#)
- [Overview: Establishing a Password Policy for an Identity Management Realm](#)

What a Password Policy Is

Password policies 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. A user account stays locked even after the lockout duration has passed unless the user binds with the correct password

If the super user account, `cn=orcladmin`, 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.

See Also:

The "oidpasswd" command-line tool reference in *Oracle Identity Management User Reference* for information on unlocking a super user account

["Possible Password Policy Problems"](#) on page J-7

- A minimum password length of five characters with at least one numeric character

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.

Note: Oracle Identity Management has two distinct types of privileged user. Both privileged user accounts can be locked if certain password policies are activated.

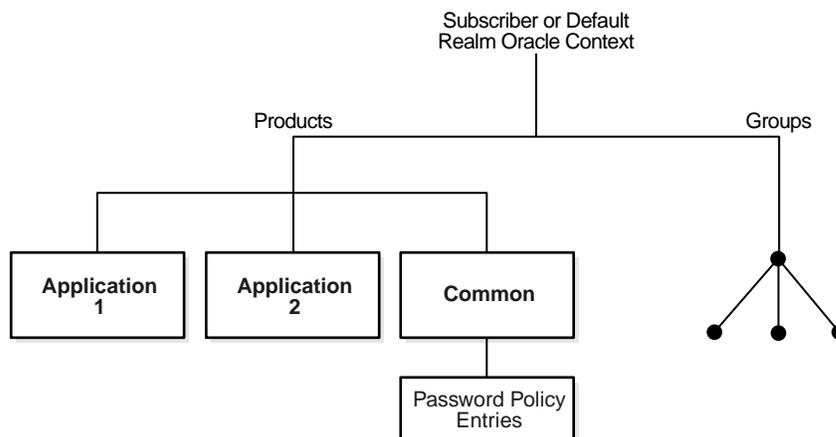
The first type of privileged user, the super user with the DN `cn=orcladmin`, is represented as a special user entry found within the default identity management realm. It enables directory administrators to make any modifications to the DIT and any changes to the configuration of Oracle Internet Directory servers. If the super user (`orcladmin`) account is locked—for example, as a result of too many attempts to bind with an incorrect password—then an administrator with DBA privileges to the Oracle Internet Directory repository can unlock it by using the `oidpasswd` tool. To unlock the `orcladmin` account execute the command:

```
oidpasswd unlock_su_acct=TRUE
```

The second privileged user, a realm-specific privileged user, governs capabilities such as creation and deletion of users and groups within a realm and all the functionality related to Oracle Delegated Administration Services. This account is represented by an entry with the DN `cn=orcladmin,cn=users, realm DN`. Note that, in contrast to the single super user account, each realm has its own realm-specific privileged user. To unlock the realm-specific privileged account, the first type of privileged user, `cn=orcladmin`, can modify the account password by using Oracle Directory Manager.

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, for example, includes:
 - 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, establishing a password policy requires doing the following:

1. Creating a password policy entry, associating it with the `pwdpolicy` object class, and populating the corresponding attributes.
2. Setting values for the `pwdPolicy` object class that contains password policy information for the entire directory. Do this during installation when the entry of this object class is created.
3. Verifying that the `orclpwdpolicyenable` attribute in the password policy entry is set to 1.

See Also: "Object Class Reference" in *Oracle Identity Management User Reference* 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 and the tools you use to perform each one, and points you to the corresponding information.

Table 15-1 Tasks and Tools for Managing Password Policies

Task	Tools	Instructions
Enabling and disabling accounts	Oracle Internet Directory Self-Service Console Idapmodify	" Enabling and Disabling Accounts by Using the Oracle Internet Directory Self-Service Console " on page 15-8 "Example: Enabling and Disabling Accounts by Using Command-Line Tools " on page 15-7
Forcing a password change	Idapmodify	"Example: Forcing a Password Change by Using Command-Line Tools " on page 15-8
Modifying password policies for an identity management realm	Oracle Directory Manager Idapmodify	" Modifying Password Policies of an Identity Management Realm by Using Oracle Directory Manager " on page 15-6 "Example: Modifying Password Policies of an Identity Management Realm by Using Command-Line Tools " on page 15-7

Table 15–1 (Cont.) Tasks and Tools for Managing Password Policies

Task	Tools	Instructions
Setting password policies	ldapmodify	" Example: Setting Password Policies by Using Command-Line Tools " on page 15-6
Unlocking accounts	Oracle Internet Directory Self-Service Console ldapmodify	" Unlocking Accounts by Using the Oracle Internet Directory Self-Service Console " on page 15-8 " Example: Unlocking Accounts by Using Command-Line Tools " on page 15-7
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 15-5 " Example: Viewing Password Policies of an Identity Management Realm by Using Command-Line Tools " on page 15-7

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 **Oracle Internet Directory Servers**, then **directory server instance**, then **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 listing the full DN of each password policy entry
- The **Password Policy Entry** column listing the corresponding RDNs of those policies

To get the latest updates to realm-specific password policies, choose **Refresh**.

To get the password policies 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 A-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 A-10](#) on page A-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 A-11](#) on page A-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 A-12](#) on page A-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 A-13](#) on page A-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](#)
- [Examples: Managing the Password Policies of an Identity Management Realm by Using Command-Line Tools](#)
- [Example: Enabling and Disabling Accounts by Using Command-Line Tools](#)
- [Example: Unlocking Accounts by Using Command-Line Tools](#)
- [Example: Forcing a Password Change by Using Command-Line Tools](#)

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
EOF

```

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 you can unlock an account without resetting the user password. This saves you from having to explicitly tell the user the new password. The user can simply log in 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
EOF
```

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 15-8 for instructions on resetting passwords

Managing Password Policies by Using the Self-Service Console

This section explains how to use the Oracle Internet Directory Self-Service Console to:

- Enable and disable accounts
- Unlock accounts
- Reset your own password

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: The section on managing accounts in *Oracle Identity Management Guide to Delegated Administration* 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: The section on managing accounts in *Oracle Identity Management Guide to Delegated Administration* 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: The section on resetting your password if you forget it in *Oracle Identity Management Guide to Delegated Administration* 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, 10g Release 2 (10.1.2), 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: ["Troubleshooting Password Policies"](#) on page J-7 for a list of the messages and information about how to resolve them

Directory Storage of Password Verifiers

Password verifiers are the security credentials used to authenticate users to Oracle components other than Oracle Internet Directory. This chapter explains how Oracle Internet Directory centrally stores these password verifiers.

This chapter contains these topics:

- [About Centralized Storage of User Authentication Credentials](#)
- [Storing and Managing Password Verifiers for Authenticating to Oracle Internet Directory](#)
- [Storing and Managing Password Verifiers for Authenticating to Oracle Components](#)
- [Verifier Generation Using Dynamic Parameters](#)

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 to 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.

The default userPassword hashing algorithm for Oracle Internet Directory has been changed from MD4 to SHA-1. This default scheme is in effect for new installations only. All userPassword attributes created after a new install will be one-way hashed using SHA-1, then stored in Oracle Internet Directory.

When you perform an upgrade, the default hashing scheme in effect prior to upgrade is retained. For example, if the default scheme prior to the upgrade was MD4, then MD4 remains the default scheme after the upgrade. To ensure greater security of userPasswords, change the default scheme to SHA-1 immediately after the upgrade. When you change the default scheme to SHA-1, user login is unaffected. For greater security, require users to reset their passwords so that SHA-1 values hash values are stored in Oracle Internet Directory.

Oracle Internet Directory stores the user password in a reversible encrypted format in an operational attribute called orclrePWD. This attribute is generated only if the attribute orclpWdEncryptionEnable in the password policy entry is set to 1. The orclrePWD 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](#)
- [Hashing Schemes for Creating Password Verifiers](#)
- [Managing Password Protection by Using Oracle Directory Manager](#)
- [Managing Password Protection by Using ldapmodify](#)

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 DSE attribute userPassword. 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.

For external users, Oracle Internet Directory generates the attribute orclrePWD during authentication. In particular this attribute is generated when clients authenticate using ldapcompare on the user's cleartext password. If the attribute orclrePWD does not exist, then the Oracle Internet Directory server generates this attribute using the cleartext password provided for authentication. However this attribute is not generated if external users are authenticated against Oracle Internet Directory using ldapbind

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 12-6

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](#)
- [Attributes for Storing Password Verifiers](#)
- [Default Verifiers for Oracle Components](#)
- [Example: How Password Verification Works for an Oracle Component](#)
- [Managing Password Verifier Profiles for Oracle Components by Using Oracle Directory Manager](#)
- [Managing Password Verifier Profiles for Oracle Components by Using Command-Line Tools](#)

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.

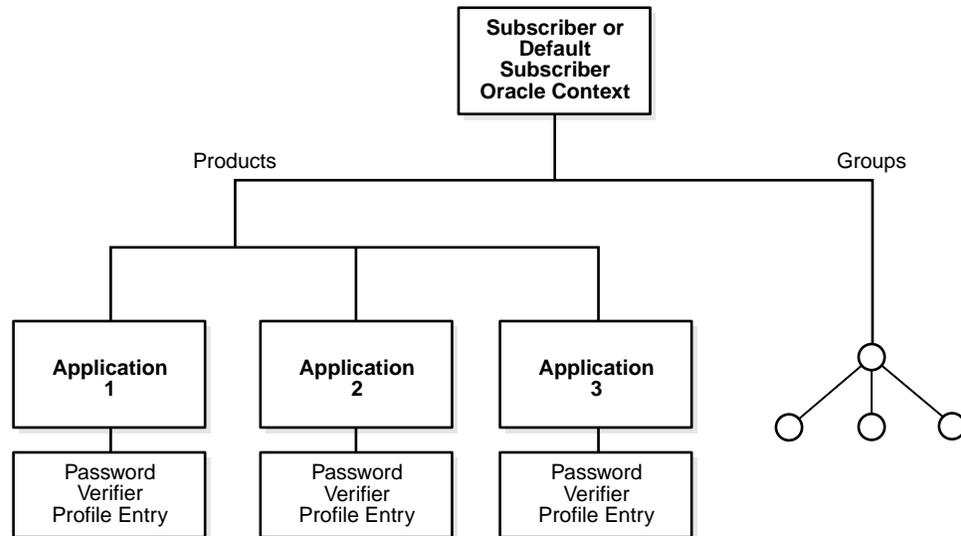
You can choose one of these hashing algorithms to derive a password verifier:

- **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](#), 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
<code>authPassword</code>	<p>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, <code>userpassword</code>. The value in this attribute is synchronized with that in the <code>userpassword</code> attribute.</p> <p>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.</p> <p>This attribute is multivalued and can contain all the other verifiers that different applications use for this user's clear text password. If the <code>userpassword</code> attribute is modified, then the <code>authpasswords</code> for all applications are regenerated.</p>
<code>orclPasswordVerifier</code>	<p>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, <code>userpassword</code>. The value in this attribute is not synchronized with that in the <code>userpassword</code> attribute.</p> <p>Like <code>authPassword</code>, this attribute is multivalued and can contain all the other verifiers that different applications use for this user's clear text password.</p>

Table 16–1 (Cont.) Attributes for Storing Password Verifiers in User Entries

Attribute	Description
orclPassword	<p>Attribute for storing only the 03LOGON verifier for enterprise users. The 03LOGON verifier is synchronized with the <code>userpassword</code> attribute, and it is generated by default for all user entries associated with the <code>orcluserv2</code> object class.</p> <p>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 <code>orcluserv2</code> object class.</p>

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 16-7, 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.

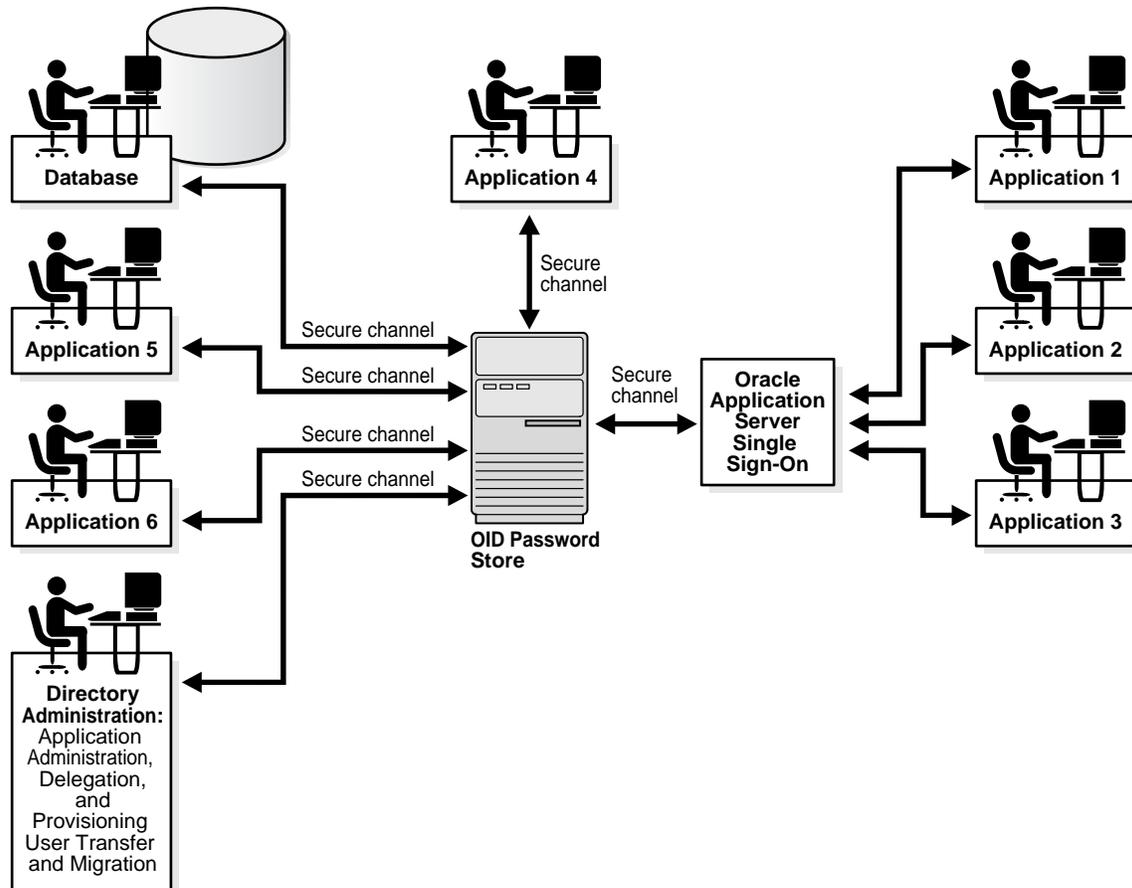
The following is an example of an application-specific verifier profile entry. Any application that chooses not to use the common verifier framework must create its own verifier profile entry, similar to the one given in the following example. The `orclappid` will be set to the GUID of the application container and it will also be used as a subtype in the verifier attributes `authpassword` and `orclpasswordverifier`.

```
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 OracleAS Unified Messaging—are stored in the `orclpasswordverifier;orclcommonpin` attribute. The subtype `orclcommonpin` is used to distinguish numeric PINs from alphanumeric passwords. Any application that uses numeric PINs can directly query or compare against the attribute `orclpasswordverifier;orclcommonpin`.

The verifiers for alphanumeric password-based applications—for example, Oracle Internet File System—can be stored in either:

- The `authpassword;orclcommonpwd` attribute—If an application requires its verifier to be synchronized with the `userpassword` attribute
- The `orclpasswordverifier;orclcommonpwd` attribute—If synchronization with the `userpassword` attribute is not required

The subtype `orclcommonpwd` is used to distinguish alphanumeric passwords from numeric PINs. The verifier attributes subtyped by `orclcommonpwd` can be queried against.

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.

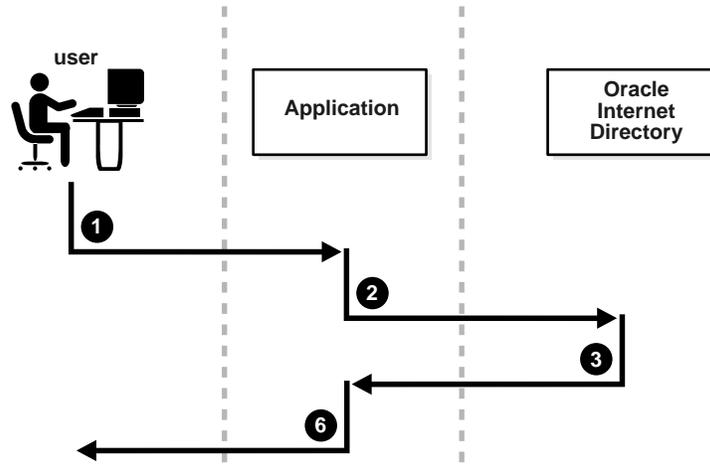
```
Cn=defaultSharedPwdProfileEntry, cn=common, cn=products, cn=oraclecontext
Objectclass: orclpwdverifierprofile
Cn: orclcommonpwdprofileentry
Orclappid: orclcommonpwd
Orclpwdverifierparams;authpassword: crypto:SASL/MD5 $ realm:Authorized_Users
Orclpwdverifierparams;authpassword: crypto:ORCLWEBDAV $ 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_application_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 **Oracle Internet Directory Servers**, then *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 A-14](#) on page A-8.
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
```

Verifier Generation Using Dynamic Parameters

The password verifiers described previously are static password verifiers. That is, they are generated with preconfigured parameters, typically during application installation. Some applications, including Oracle Calendar, Oracle Email, and Oracle Wireless and Voice, require Oracle Internet Directory to generate dynamic password verifiers.

This section contains these topics:

- [Generating Dynamic Password Verifiers](#)
- [Configuring Oracle Internet Directory to Generate Dynamic Password Verifiers](#)

Generating Dynamic Password Verifiers

Oracle Internet Directory generates a dynamic password verifier when an application requests one. Dynamic verifiers are based on application parameters that are not available until run time.

In order to generate a dynamic password verifier, Oracle Internet Directory needs a user password that was previously stored in a reversible encrypted format. Oracle Internet Directory stores such values in the operational attributes `orclrevpwd` and `orclunsyncrevpwd`. Encrypted values based on `userpassword` are stored in the parameter `orclrevpwd`. Encrypted values based on passwords other than `userpassword`, such as the numeric PINs used by Oracle Calendar, are stored in the parameter `orclunsyncrevpwd`.

Configuring Oracle Internet Directory to Generate Dynamic Password Verifiers

If you are deploying applications that use `userpassword` and that need dynamic password verifiers, you must ensure that Oracle Internet Directory generates the `orclrevpwd` parameter. Oracle Internet Directory generates the attribute `orclrevpwd` when you provision a user if the attribute `orclpwdencryptionenable` in the realm password policy entry is set to 1. Therefore, you must set `orclpwdencryptionenable` to 1 before you provision users. Alternatively, if users were provisioned before you set `orclpwdencryptionenable`, all users must reset their user passwords to trigger the generation of the encrypted value.

If you are deploying applications that use a numeric PIN and that need dynamic password verifiers, you must ensure that Oracle Internet Directory can use the crypto type 3DES in order to generate the value stored in `orclunsyncrevpwd`. You must specify 3DES as a value of the attribute `orclpwdverifierparams;orclpasswordverifier` in the common verifier profile entry under the root oracle context. The default DN of this entry is `cn=DefaultSharedPINProfileEntry,cn=Common,cn=Products,cn=OracleContext`. To set the value, you would specify:

```
dn: cn=DefaultSharedPinProfileEntry, cn=Common,
    cn=Products, cn=Oraclecontext
cn: DefaultSharedPinProfileEntry
orclappid: orclcommonpin
orclpwdverifierparams;orclpasswordverifier: crypto:MD5
orclpwdverifierparams;orclpasswordverifier: crypto:3DES
```

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](#)
- [Overview: Privileges for Administering the Oracle Technology Stack](#)
- [Delegation of Privileges for User and Group Management](#)
- [Delegation of Privileges for Deployment of Oracle Components](#)
- [Delegation of Privileges for Component Runtime](#)

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](#)
- [Delegation in an Oracle Application Server Environment](#)
- [About the Default Configuration](#)
- [Overview: Privileges for Administering the Oracle Technology Stack](#)

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.

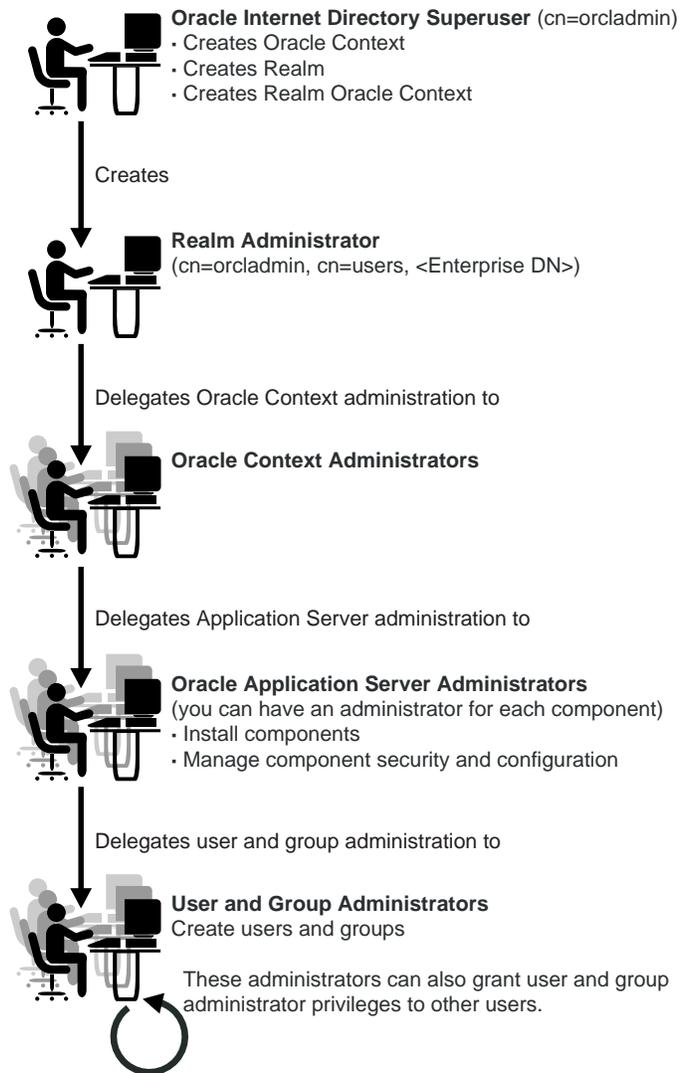
To delegate the necessary privileges, you assign the user to the appropriate administrative group. For example, suppose that you store data for both enterprise users and the e-mail service in the directory, and need to specify a unique administrator for each set of data. To specify a user as the administrator 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 17-2 shows, in an Oracle Application Server environment the directory super user (cn=orcladmin) creates:

- The Oracle Context
- The realm
- The realm-specific Oracle Context
- The entry for the realm administrator (cn=orcladmin, cn=users, Enterprise DN)

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. The Oracle Application Server Administrators install and administer Oracle Application Server components and delegate administration of user and group data to the User and Group Administrators group. The User and Group Administrators create users and groups. They can also grant user and group administrator privileges to other users.

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 17-2](#).

Table 17-1 *Default Privileges Granted to Everyone and to Each User*

Subject	Default Privileges
Everyone	The following privileges at the Root DSE: <ul style="list-style-type: none"> ■ Permission to browse user entries ■ Search, read, and compare access for all user attributes except the following <code>userpkcs12</code>, <code>orcluserpkcs12hint</code>, <code>userpassword</code>, <code>orclpassword</code>, and <code>orclpasswordverifier</code>
Each user	Complete access to his or her own attributes—including the <code>userpassword</code> , <code>orclpassword</code> , and <code>orclpasswordverifier</code> 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

Type of Privilege	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 17-4
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 17-8
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 17-10

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: ["The Default Directory Structure of Oracle Internet Directory"](#) on page 23-1 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](#)
- [Default Privileges for Managing User Data](#)
- [Default Privileges for Managing Group Data](#)

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	<code>cn=oracleDASCreateUser, cn=groups, Oracle_Context_DN.</code>

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.
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	<code>cn=oracleDASEditUser,cn=groups,Oracle_Context_DN</code>

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	<code>cn=oracleDASDeleteUser,cn=groups,Oracle_Context_DN</code>

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	<code>cn=oracleDASUserPriv,cn=groups,Oracle_Context_DN</code>

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
DN	<code>cn=oracleDASCreateGroup,cn=groups,Oracle_Context_DN</code>

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	<code>cn=oracleDASEditGroup,cn=groups,Oracle_Context_DN</code>

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
DN	<code>cn=oracleDASDeleteGroup, cn=groups, Oracle_Context_DN</code>

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	<code>cn=oracleDASUserPriv, cn=groups, Oracle_Context_DN</code>

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	<p>Perform repository database installation that creates a repository database registration entry in the directory</p> <p>Perform mid-tier installation. To associate a mid-tier with a repository, the user must have the appropriate privileges with a specific repository database.</p> <p>Install and configure Oracle Application Server components that create application entities in Oracle Internet Directory</p> <p>Grant to component entities the runtime privileges listed later in this section</p> <p>Configure provisioning profiles for components so that the components can receive update notifications</p>
Privileges this group can delegate to components	<p>Read Common User Attributes—except passwords, certificates, and similar security credentials</p> <p>Read common group attributes</p> <p>Create, edit, and delete groups</p> <p>Authenticate a user</p> <p>Read application verifiers</p>
Administrators	<p>Oracle Internet Directory super user</p> <p>Oracle Context Administrator</p> <p>Owners of this group</p>
DN	<code>cn=IASAdmins,cn=groups,Oracle_Context_DN</code>

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	<code>cn=IAS & User Mgmt Admins,cn=groups,Oracle_Context_DN</code>

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
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	<code>cn=Trusted Application Admins,cn=groups,Oracle_Context_DN</code>

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 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 the Service Registry](#)
- [Default Privileges for Administering the Service Registry](#)

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 <code>userpkcs12</code> , <code>orclpkcs12hint</code> , <code>userpassword</code> , <code>orclpassword</code> , and <code>orclpasswordverifier</code> 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
DN	<code>cn=oracleUserSecurityAdmins,cn=groups,Oracle_Context_DN</code>

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 <code>userPassword</code> 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	<code>cn=authenticationServices,cn=groups,Oracle_Context_DN</code>

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	<code>cn=verifierServices,cn=groups,Oracle_Context_DN</code>

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 <code>owner</code> attribute in the group or members of the Oracle Application Server Administrators Group. Members of the Trusted Application Administrators Group
DN	<code>cn=userProxyPrivilege,cn=groups,OracleContextDN</code>

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	<code>cn=oracleContextAdmins,cn=groups,Oracle_Context_DN</code>

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	<code>cn=commonuserattributes,cn=users,Oracle_Context_DN</code>

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 17-14.

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: <code>cn</code> , <code>uniquemember</code> , <code>displayname</code> , and <code>description</code> .
Administrators	The Oracle Internet Directory super user Members of the Application Server Administrators Group Owners of this group
DN	<code>cn=commongroupattributes,cn=groups,Oracle_Context_DN</code>

Default Privileges for Reading the Service Registry

To view the contents of the Service Registry requires membership in the Service Registry Viewers Group described in [Table 17–21](#) on page 17-14.

Table 17–21 Characteristics of the Service Registry Viewers Group

Characteristic	Description
Default ACP	The default ACL is on the Services container in the root Oracle Context.
Administrators	The Oracle Internet Directory super user Members of the Application Server Administrators Group Owners of this group
DN	<code>cn=service registry viewers,cn=services,cn=rootoraclecontext,</code>

Default Privileges for Administering the Service Registry

To administer the Service Registry requires membership in the Service Registry Administrators Group described in [Table 17–22](#) on page 17-14.

Table 17–22 Characteristics of the Common Group Attributes Group

Characteristic	Description
Default ACP	The default ACL is on the Services container in the root Oracle Context.

Table 17-22 (Cont.) Characteristics of the Common Group Attributes Group

Characteristic	Description
Administrators	The Oracle Internet Directory super user Members of the Application Server Administrators Group Owners of this group
DN	<code>cn=service_registry</code> <code>admins,cn=services,cn=rootoraclecontext,</code>

Part IV

Directory Deployment

This part discusses important deployment considerations. It includes these chapters:

- [Chapter 18, "Directory Deployment Considerations"](#)
- [Chapter 19, "Deployment of Oracle Identity Management Realms"](#)
- [Chapter 20, "Capacity Planning for the Directory"](#)
- [Chapter 21, "Tuning Considerations for the Directory"](#)
- [Chapter 22, "Garbage Collection in Oracle Internet Directory"](#)
- [Chapter 23, "Migration of Data from Other Data Repositories"](#)

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](#)
- [Oracle Directory Integration and Provisioning](#)
- [Capacity Planning, Sizing, and Tuning](#)

See Also:

- [Chapter 20, "Capacity Planning for the Directory"](#) for more detailed information about capacity planning
- "High Availability and Failover Considerations" in *Oracle Application Server High Availability Guide* for more detailed information about high availability
- [Chapter 21, "Tuning Considerations for the Directory"](#) for more detailed information about tuning
- *Oracle Application Server High Availability Guide* 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. They do this 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 19-1

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 2-16

This section contains these topics:

- [An Ideal Deployment](#)
- [Partitioning Considerations](#)
- [Replication Considerations](#)
- [High Availability Considerations](#)

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 18-4 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 **Oracle Database 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 the Oracle Database, other failover technologies, such as Oracle Real Application Clusters, are also available.

See Also:

- "High Availability and Failover Considerations" in *Oracle Application Server High Availability Guide* 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

Oracle Directory Integration and Provisioning

You can reduce administrative time and costs by integrating your applications and directories—including third-party LDAP directories—with Oracle Internet Directory. Directory Integration and Provisioning, 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. Directory Integration and Provisioning 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. Directory Integration and Provisioning provides this notification through the Oracle Directory Provisioning Integration Service.

Throughout the integration process, the Directory Integration and Provisioning 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 Directory Integration and Provisioning can synchronize the data in Microsoft Active Directory with that in Oracle Internet Directory.

See Also: *Oracle Identity Management Integration Guide*

Capacity Planning, Sizing, and Tuning

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

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

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

- **Planning phase**
During this phase, gather the requirements of all directory users and establish a unified performance and capacity requirement. This consists of capacity planning and system sizing.
- **Implementation phase**
Once you have the hardware, tune the Oracle Internet Directory software stack for best use of the hardware resources. This improves the performance of Oracle Internet Directory and of the LDAP client applications.

This section contains these topics:

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

Capacity Planning

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

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

- The type of LDAP client applications
- The number of users accessing those applications
- The nature of LDAP operations those applications perform
- The number of entries in the DIT
- The type of operations performed against the Oracle directory server
- The number of concurrent connections to the Oracle directory server
- The peak rate at which operations need to be performed by the Oracle directory server
- The average latency of operations required under peak load conditions

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

Sizing Considerations

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

- The type and number of CPUs for the Oracle Internet Directory server computer
- The type and size of disk subsystems for the Oracle Internet Directory server computer

- The amount of memory required for the Oracle Internet Directory server computer
- The type of network used for LDAP messages from the clients

Based on current experience, [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 18–2](#) gives 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 18-7 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

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:

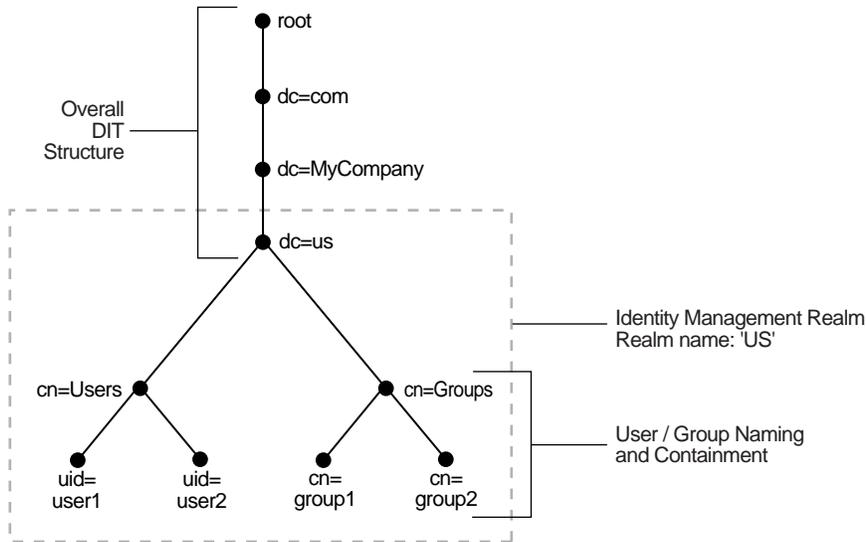
- [Planning the Directory Information Tree for Identity Management](#)
- [Identity Management Realms in an Enterprise Deployment](#)
- [Identity Management Realms in a Hosted Deployment](#)
- [Identity Management Realm Implementation in Oracle Internet Directory](#)
- [Default Directory Information Tree and the Identity Management Realm](#)
- [Administration of Identity Management Realms](#)

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
- An efficient physical deployment of the directory service
- Easier configuration of synchronization of a third-party directory with Oracle Internet Directory

[Figure 19-1](#) shows a directory information tree for a hypothetical company, called MyCompany, that is deploying identity management.

Figure 19–1 Planning the Directory Information Tree


MyCompany 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=mycompany`, `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.

Planning the logical organization of the directory for Oracle Identity Management involves planning:

- The overall structure of the directory information tree
- The directory containment and naming for users and groups
- The identity management 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](#)
- [Migrating a DIT from a Third-Party Directory](#)

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 either full or partial replication is planned, then proper boundaries and policies for 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 `mycompany.com`, then a directory structure that has `dc=mycompany,dc=com` is recommended. Oracle Corporation recommends that you not use departmental or organization level domain components such as `engineering.in.engineering.mycompany.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=mycompany,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`.
- The default attribute for uniquely identifying users is `cn` or `CommonName`. The typical value of `CommonName` is the person's full name. People's names or email addresses, however, can change and therefore might not be suitable as values of this attribute. If possible choose a value that will not change and that uniquely identifies a user, such as the employee id.
- 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 in Oracle Internet Directory with the one used in the third-party directory. This simplifies the synchronization and 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

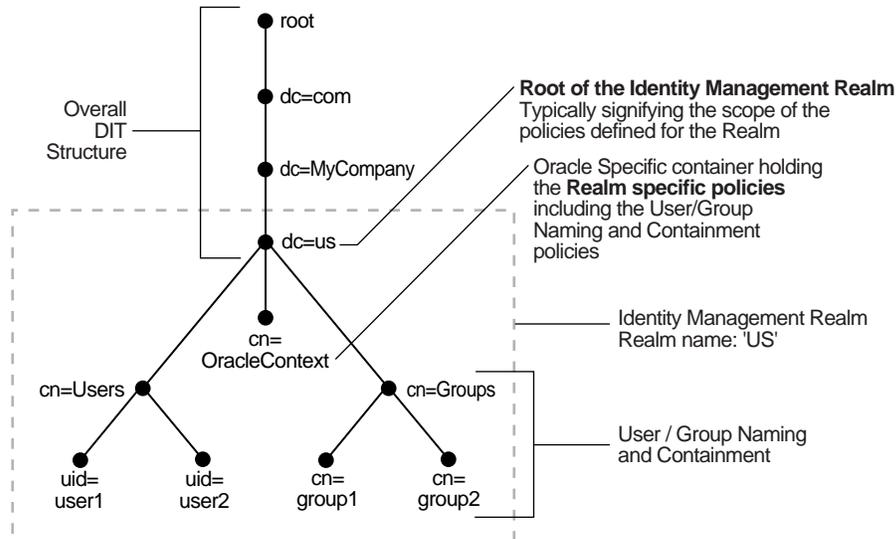
Once you have decided on the overall DIT structure and the placement of users and groups, you need to identify the directory entry to serve as the root of the identity management realm. This entry determines the scope of the identity management policies defined in the realm. By default, the scope is 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 19-2](#) shows an example of an identity management realm for an enterprise called MyCompany.

Figure 19–2 Example of an Identity Management Realm

In the example in [Figure 19–2](#), the deployment uses a domain name-based DIT structure. The container `dc=us`, `dc=mycompany`, `dc=com` is 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`.

Migrating a DIT from a Third-Party Directory

To migrate a DIT from a third-party directory, use the techniques described in *Oracle Identity Management Integration Guide* for synchronizing with third-party metadirectory solutions and integrating with third-party directories. If you are migrating a DIT from a Microsoft Active Directory environment, also see the chapter on integration with the Microsoft Active Directory Environment. Oracle recommends that you configure the Oracle Internet Directory DIT to be identical to the third-party DIT.

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:

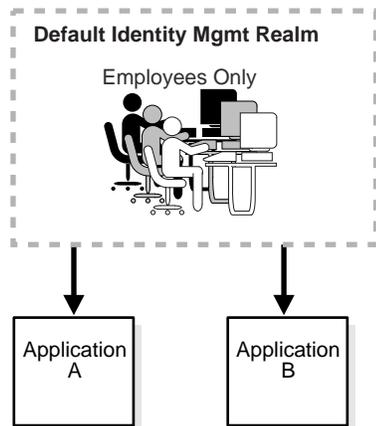
- [Single Identity Management Realm in the Enterprise](#)
- [Multiple Identity Management Realms in the Enterprise](#)

Single Identity Management Realm in the Enterprise

In this scenario, an enterprise has a single set of users, all of whom are managed with the same identity management policies. This is the default configuration of all Oracle products. It includes only one default identity management realm in Oracle Internet Directory and all Oracle components in the enterprise serve users in that realm.

[Figure 19–3](#) illustrates this usage.

Figure 19–3 Enterprise Use Case: Single Identity Management Realm

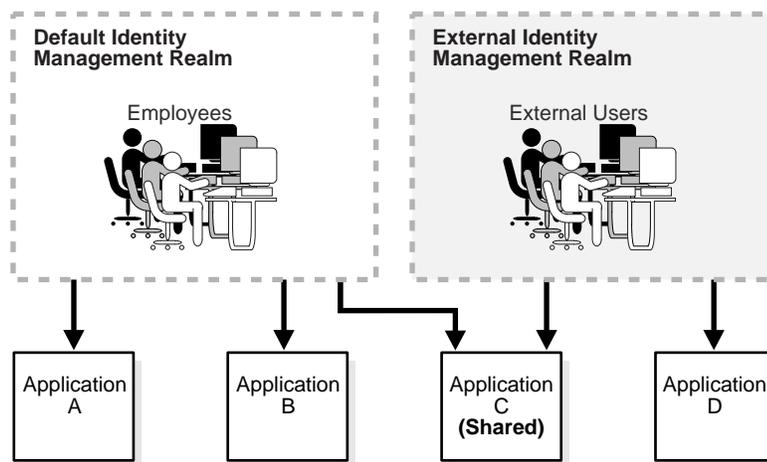


In the example in [Figure 19–3](#), there is a single, default identity management realm containing employees only. In that realm all users and groups are managed and all share access to the same applications, Application A and Application B.

Multiple Identity Management Realms in the Enterprise

You 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, you can deploy two realms, one for internal and one for external users. [Figure 19–4](#) on page 19-8 illustrates this usage.

Figure 19–4 Enterprise Use Case: Multiple Identity Management Realms



In the example in [Figure 19–4](#), 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.

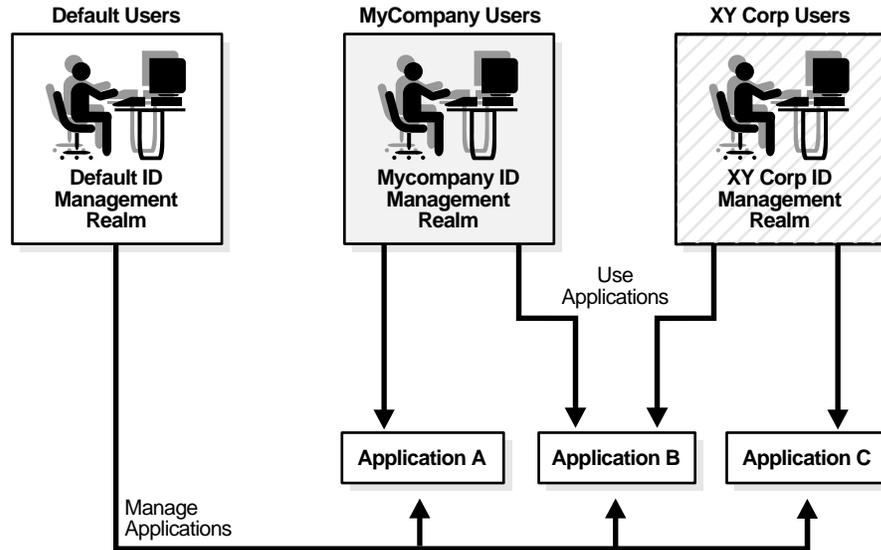
Identity Management Realms in a Hosted Deployment

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 19-5 shows a hosted deployment with two hosted companies.

Figure 19-5 Hosted Deployment Use Case



In the example in Figure 19-5, the ASP users are in the default identity management realm. The ASP manages its users, groups and associated policies in that realm. ASP users manage Applications A, B, and C for the hosted companies. Hosted company MyCompany users are in the Mycompany identity management realm. They use Applications A and B. Hosted company XY Corp users are in the XY Corp identity management realm. They use Applications B and C.

Identity Management Realm Implementation in Oracle Internet Directory

Table 19-1 describes the objects in an identity management realm.

Table 19-1 Oracle Identity Management Objects

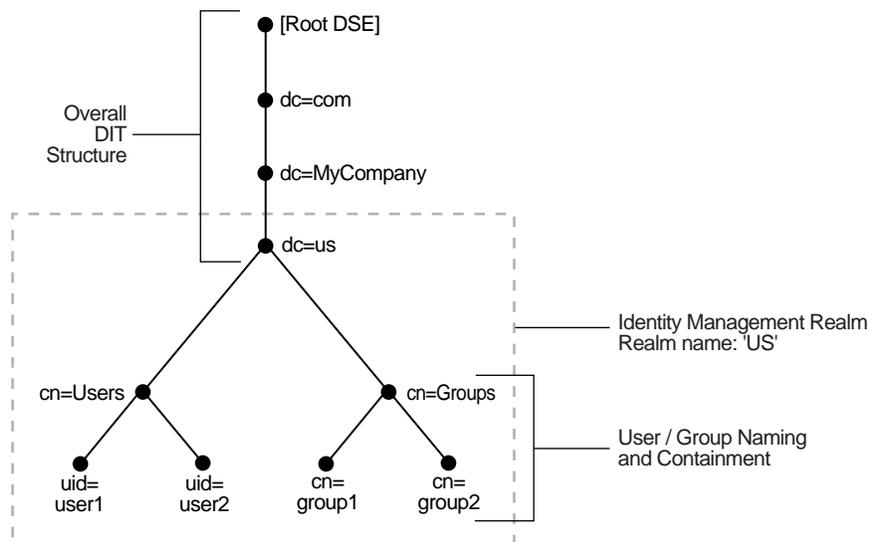
Object	Description
Root Oracle Context	This object contains: <ul style="list-style-type: none"> ▪ A pointer to the default identity management realm in the infrastructure ▪ Information on how to locate a realm given a simple name of the realm
Identity Management Realm	A normal directory entry with a special object class associated with it.

Table 19–1 (Cont.) Oracle Identity Management Objects

Object	Description
Identity Management Realm-Specific Oracle Context	<p>In each realm, this object contains:</p> <ul style="list-style-type: none"> ■ 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

Default Directory Information Tree and the Identity Management Realm

To make configuration easier, Oracle Internet Directory, at installation, creates a default DIT and sets up a default identity management realm.

Figure 19–6 Default Identity Management Realm

As [Figure 19–6](#) shows, the default identity management realm is part of a global DIT. The node just below the root DSE is `dc=com`, followed by `dc=MyCompany`, then `dc=us`. These four nodes represent the overall DIT structure. The node `dc=us` is the root of the default identity management realm. It has two subtrees for containing user and group information: `cn=Users` and `cn=Groups`. For purposes of illustration, the `cn=Users` node contains two leaves: `uid=user1` and `uid=user2`. Similarly, the `cn=Groups` node contains `cn=group1` and `cn=group2`.

Oracle Internet Directory 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`.

It also gives you the option of specifying a different DN that meets your deployment needs as the root of your default identity management realm on the **Specify Namespace in Internet Directory** install screen. For example, if you plan to integrate

your Identity Management installation with a third-party directory, it is recommended that you specify a DN that matches the DN of the default naming context in the third-party directory. For more details on obtaining the default naming context from a third-party directory, refer to the chapter on integrating with third-party directories in *the Oracle Identity Management Integration Guide*.

During configuration, 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 is located under the users container. In this example, the fully qualified DN of the realm administrator is `orcladmin,cn=users,dc=us,dc=mycompany,dc=com`.

Note: If the realm administrator's account becomes locked, the Oracle Internet Directory super user can unlock it by modifying the realm administrator's account password, using Oracle Directory Manager.

- 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 realm administrator 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:

- "Oracle Identity Management LDAP Object Classes" in *Oracle Identity Management User Reference* 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 the Default Identity Management Realm](#)
- [Creating Additional Identity Management Realms for Hosted Deployments](#)

Customizing the Default 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 the Default Identity Management Realm

What You Can Customize	Tools	Information
Directory structure and naming policies	Oracle Internet Directory Self-Service Console Oracle Directory Manager Command-line tools	"Changing the Location of Users and Groups In The Default Identity Management Realm" on page 19-13 "Planning the Directory Information Tree for Identity Management" on page 19-1 The chapter on using the Oracle Internet Directory Self-Service Console in <i>Oracle Identity Management Guide to Delegated Administration</i>
Authentication policies	Oracle Directory Manager Command-line tools	Chapter 15, "Password Policies in Oracle Internet Directory"
Identity management authorizations	Oracle Internet Directory Self-Service Console Oracle Directory Manager Command-line tools	Chapter 17, "Delegation of Privileges for an Oracle Technology Deployment" The chapter on using the Oracle Internet Directory Self-Service Console in <i>Oracle Identity Management Guide to Delegated Administration</i>

Changing the Location of Users and Groups In The Default Identity Management Realm

A typical scenario where this might be required is one where you need to integrate your Oracle Identity Management installation with a third-party directory.

For example, assume the default Identity Management Realm is `dc=mycompany,dc=com` and there are users under `cn=users,dc=mycompany,dc=com`.

If the third party directory naming context does not match the current user and group search base in the default realm, then you can alter the user and group search base of the default realm so that both the existing users and the third party users can login using SSO. Select a user search base just high enough to include the existing users and the third party users. Let us call this search base the Lowest Common User Search Base.

Note: This approach assumes that the user `nickname` attribute selected for SSO Login is unique across the existing user search base and the third party directory naming context. Otherwise, SSO authentication will fail for all those users whose `nickname` attribute values clash.

If your deployment scenario matches any of the use cases from 1 to 5, follow the procedure described in ["Steps to Update the Existing User and Group Search Base"](#) on page 19-14.

Use Case 1:

The third party naming context is under the default realm, but in a different container than the realm user search base

For example, the existing users are under `cn=users,dc=mycompany,dc=com` and the third party naming context is under `cn=users,o=employees,dc=mycompany,dc=com`. In this case, the Lowest Common User Search Base is `dc=mycompany,dc=com`.

Use Case 2:

The third party naming context is outside the default realm, but there is a Lowest Common User Search Base.

For example, the existing users are under `cn=users,dc=mycompany,dc=com` and the third party naming context is under `cn=users,dc=mycompanyecorp,dc=com`. In this case, the Lowest Common User Search Base is `dc=com`.

If the Lowest Common User Search Base is the root DSE, then use the procedures described for [Use Case 6](#):

1. ["Set up an Additional Search Base"](#) on page 19-15.
2. ["Refresh SSO"](#) on page 19-16.
3. ["Reconfigure Provisioning Profiles"](#) on page 19-17.

Use Case 3:

The third party naming context is the same as the default realm DN.

For example, the existing users are under `cn=users,dc=mycompany,dc=com` and the third party naming context is directly under `dc=mycompany,dc=com`. In this case, the Lowest Common User Search Base is `dc=mycompany,dc=com`.

Use Case 4:

The third party naming context contains the parent of the default realm DN.

For example, you might have a default realm with DN:
`dc=us,dc=mycompany,dc=com`, existing users under
`cn=users,dc=us,dc=mycompany,dc=com` and the third party naming context
directly under `dc=com`. In this case, the Lowest Common User Search Base is `dc=com`.

Use Case 5:

The third party naming context is under the existing user search base.

For example, existing users are under `cn=users,dc=mycompany,dc=com` and the third party naming context is directly under
`l=emea,dc=mycompany,dc=com`. In this case, the Lowest Common User Search Base is `cn=users,dc=mycompany,dc=com`. In this use case, you do not need to change the user search base.

Steps to Update the Existing User and Group Search Base You must perform the following steps before you set up synchronization with the third party directory.

1. Back up the Oracle Internet Directory database.
2. Create the user and group containers for the third party directory, using Oracle Directory Manager, if the entries do not already exist in the directory.
3. Apply appropriate ACLs on the new users container by doing the following:
 - a. Instantiate the variables `%USERBASE%` and `%REALMBASE%` in the ACL template file `$ORACLE_HOME/ldap/schema/oid/oidUserAdminACL.sbs` and create the file `usracl.ldif`. Set the variable `%USERBASE%` to the DN of the new user container and the variable `%REALMBASE%` to the default realm DN.
 - b. Upload the instantiated LDIF file `usracl.ldif` using the `ldapmodify` command.
4. Apply appropriate ACLs on the new groups container by doing the following:
 - a. Instantiate the variables `%GRPBASE%` and `%REALMBASE%` in the ACL template file `$ORACLE_HOME/ldap/schema/oid/oidGroupAdminACL.sbs` and create the file `grpac1.ldif`. Set the variable `%USERBASE%` to the DN of the new user container and the variable `%REALMBASE%` to the default realm DN.
 - b. Upload the instantiated LDIF file `grpac1.ldif` using the `ldapmodify` command.
5. Determine a Lowest Common User Search Base base that is just high enough to include the existing users and the third party users.

For example, if existing users are under `cn=users,dc=mycompany,dc=com` and the third party users are under `l=emea,dc=mycompany,dc=com`, then the Lowest Common User Search Base is `dc=mycompany,dc=com`.

The Lowest Common User Search Base might be the root entry. That is the case if, for example, the existing users are under `cn=users,dc=mycompany,dc=com` and the third party users are under `dc=mycompanycorp,dc=net`. In that case,

skip to the deployment scenario described in Use Case 6: "[Set up an Additional Search Base](#)" on page 19-15

6. If you must also synchronize groups, determine a group search base that is just high enough to include the existing groups and the third party groups. Lets call this search base the Lowest Common Group Search Base.

For example, if existing groups are under `cn=groups,dc=mycompany,dc=com` and the third party groups are under `l=emea,dc=mycompany,dc=com`, then the Lowest Common Group Search Base is `dc=mycompany,dc=com`.
7. Log into the Self-Service Console as the administrator of the realm (usually `orcladmin`).
8. Go to the **Configuration** tab and set the user search base to the Lowest Common User Search Base you determined in step 5. If you must also synchronize groups, then also then set the group search base to the Lowest Common Group Search Base that you determined in step 6.
9. To make SSO recognize these changes, follow the procedure described under "[Refresh SSO](#)" on page 19-16.
10. Verify the SSO login of users in the original user search base by logging in as `orcladmin`.
11. You must also reconfigure the applications that have been provisioned to reflect the modified user and group bases. Follow the steps described under "[Reconfigure Provisioning Profiles](#)" on page 19-17.

Note: In addition to the user and group search base attributes, you can also modify other configuration settings of an identity management realm, such as the attribute for Login Name (nickname) or the attribute for RDN, using the Self-Service Console. See: "Modifying Configuration Settings for an Identity Management Realm" in the chapter "Using the Oracle Internet Directory Self-Service Console" of *Oracle Identity Management Guide to Delegated Administration* for more details.

Use Case 6:

In this case, the third party naming context is outside the default realm and the Lowest Common User Search Base is the root DSE.

For example, if existing users are under `cn=users,dc=mycompany,dc=com` and the third party naming context is under `cn=users,dc=mycompanycorp,dc=net`, then the Lowest Common User Search Base is the root DSE.

In this case, you must add the third party naming context as an additional search base. The steps are as follows:

1. "[Set up an Additional Search Base](#)"
2. "[Refresh SSO](#)"
3. "[Reconfigure Provisioning Profiles](#)"

Set up an Additional Search Base Perform the following steps before setting up synchronization with the third party directory.

1. Back up the Oracle Internet Directory database.

2. Create the user and group containers for the third party directory, using Oracle Directory Manager, if the entries do not already exist in the directory.
3. Apply appropriate ACLs on the new users container by doing the following:
 - a. Instantiate the variables `%USERBASE%` and `%REALMBASE%` in the ACL template file `$ORACLE_HOME/ldap/schema/oid/oidUserAdminACL.sbs` and create the file `usracl.ldif`. Set the variable `%USERBASE%` to the DN of the new user container and the variable `%REALMBASE%` to the default realm DN.
 - b. Upload the instantiated LDIF file `usracl.ldif` using the `ldapmodify` command.
4. Apply appropriate ACLs on the new groups container by doing the following:
 - a. Instantiate the variables `%GRPBASE%` and `%REALMBASE%` in the ACL template file `$ORACLE_HOME/ldap/schema/oid/oidGroupAdminACL.sbs` and create the file `grpACL.ldif`. Set the variable `%USERBASE%` to the DN of the new user container and the variable `%REALMBASE%` to the default realm DN.
 - b. Upload the instantiated LDIF file `grpACL.ldif` using the `ldapmodify` command.
5. Log into the Self-Service Console as the administrator of the realm.
6. Go the **Configuration** tab.
 - a. Add `cn=users,dc=mycompanycorp,dc=net` to the `usersearchbase` for the current realm.
 - b. Add `cn=groups,dc=mycompanycorp,dc=net` to the `groupsearchbase` for the current realm.
7. To make Oracle Application Server Single Sign-On recognize these changes, follow the procedure described under "[Refresh SSO](#)" on page 19-16.
8. Verify the SSO login of users in the original user search base by logging in as `orcladmin`.
9. If mid-tiers have been configured against this identity management configuration, then you must also reconfigure the applications that have been provisioned to reflect the modified user and group bases. Follow the steps described under "[Reconfigure Provisioning Profiles](#)" on page 19-17.

Note: In addition to the user and group search base attributes, you can also modify other configuration settings of an identity management realm, such as the attribute for Login Name (nickname) or the attribute for RDN, using the Self-Service Console. See: "Modifying Configuration Settings for an Identity Management Realm" in the chapter "Using the Oracle Internet Directory Self-Service Console" of *Oracle Identity Management Guide to Delegated Administration* for more details.

Refresh SSO The steps for making Oracle Application Server Single Sign-On recognize your configuration changes are as follows:

1. Execute the SSO refresh script by changing to the directory `$ORACLE_HOME/sso/admin/plsql/sso/` and typing:

```
sqlplus orasso/password@ssoreoid.sql
```

To get the `orasso` schema password, refer to the appendix "Obtaining the Single Sign-On Schema Password" in the *Oracle Application Server Single Sign-On Administrator's Guide*.

- Restart the `OC4J_SECURITY` instance by typing:

```
opmnctl restartproc type=oc4j instancename=OC4J_SECURITY
```

Reconfigure Provisioning Profiles If you installed middle-tier applications against this default identity management realm before changing its user and group search bases, then the provisioning profiles created by the middle-tier installations become invalid. This happens because the profiles have the old user or group search base information in the `event_subscriptions` attribute. You must modify all the profiles by using `oidprovtool`.

Execute the following steps for every provisioning profile:

- Use `ldapsearch` to put all the provisioning profile information into an LDIF file:

```
ldapsearch -h oid_host -p oid_port \
-D "cn=orcladmin" -w password -s sub \
-b "cn=provisioning profiles,cn=changelog subscriber,\
cn=oracle internet directory" \
"objectclass=*" > provprofiles.ldif
```

The event subscriptions look something like this:

```
USER:cn=users,dc=mycompany,dc=com:MODIFY(list_of_attributes)
USER:cn=users,dc=mycompany,dc=com:DELETE
GROUP:cn=groups,dc=mycompany,dc=com:MODIFY(list_of_attributes)
GROUP:cn=groups,dc=mycompany,dc=com:DELETE
```

where `cn=users,dc=mycompany,dc=com` and `cn=groups,dc=mycompany,dc=com` are the user and group search bases, respectively, that were created when you installed and configured the application.

- Get the actual DNs of the application identity by searching the Oracle Internet Directory server based on the GUID. To get the application DN, type:

```
ldapsearch -h host -p port -D cn=orcladmin -w password \
-s sub -b "" \
"orclguid=Value_of_orclODIPProvisioningAppGuid" dn
```

You can get the GUID values for each profile from the attribute values in `provprofiles.ldif`.

- Modify each of the returned profiles as follows:

```

$ORACLE_HOME/bin/oidprovtool operation=MODIFY \
ldap_host=host ldap_port=port \

ldap_user_dn="cn=orcladmin" \

ldap_user_passwd=password \

interface_version=interfaceVersion \
application_dn=applicationDN \
organization_dn=identity_Realm_DN \
event_subscription=New_Event_Subscription_1
event_subscription=New_Event_Subscription_2

```

```
event_subscription=New_Event_Subscription_n
```

The `New_Event_Subscription` arguments should be of the form:

```
USER: new_user_search_base:MODIFY(list_of_attributes)
USER: new_user_search_base:DELETE
GROUP: new_group_search_base:MODIFY(list_of_attributes)
GROUP: new_group_search_base:DELETE
```

Here, the `organization_dn` value should be the original identity realm DN

Creating Additional Identity Management Realms for Hosted Deployments

You can create additional identity management realms by using the Oracle Internet Directory Self-Service Console.

Only members of the `ASPAadmins` group can create a new identity management realm. Use Oracle Directory Manager to add a user to that group by adding the userDN to the `uniquemember` attribute of group `ASPAadmins` in the Default Identity Management Realm-specific OracleContext. Refer to the section on "Modifying a Static Group Entry by Using Oracle Directory Manager" for details.

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:

- ["Modifying a Static Group Entry by Using Oracle Directory Manager"](#) on page 9-6.
- The chapter on the Oracle Internet Directory Self-Service Console in the *Oracle Identity Management Guide to Delegated Administration* for information about creating an additional identity management realm.

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](#)
- [Getting to Know Directory Usage Patterns: A Case Study](#)
- [I/O Subsystem Requirements](#)
- [Memory Requirements](#)
- [Network Requirements](#)
- [CPU Requirements](#)
- [Summary of Capacity Plan for Acme Corporation](#)

About Capacity Planning

If Oracle Internet Directory and the corresponding Oracle 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.

To be successful, every new installation of the Oracle Internet Directory requires:

- Adequate hardware resources to satisfy user demands at peak load rates
- A well tuned system—of both hardware and software—that makes the best use of available resources and squeezes the maximum performance out of available hardware

[Table 20-1](#) defines important terms used in this chapter.

Table 20–1 Capacity Planning Terminology

Term	Definition
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.

In this chapter, we look 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 apply our recommendations to the example of Acme Corporation.

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 assumptions in [Table 20–2](#) that enable us to guess the size of the directory.

Table 20–2 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 described in [Table 20-3](#).

Table 20-3 Overall Count of Entries

Entry Type	Size
User entries	40,000 (these represent the users themselves)
Private aliases of users	$40,000 \times 10 = 400,000$ entries
Private distribution lists of users	$40,000 \times 10 = 400,000$ entries
Company wide distribution lists	4000
Company wide aliases	1000

These assumptions yield a directory population of approximately 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 an average of five recipients for each e-mail sent by a user, this would result in five directory lookups for each e-mail.

[Table 20-4](#) summarizes all the possible directory lookups that can happen in one day.

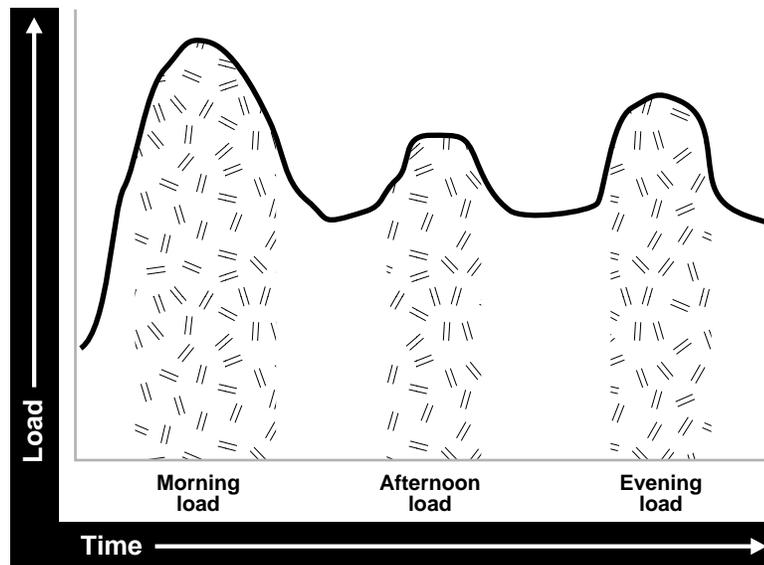
Table 20-4 Directory Lookups in a Single Day

Type of Directory Lookup	Number of Directory Lookups In One Day
The Mail Transfer Agent (MTA) processing outbound mail from each user	$5 \times 10 \times 40,000 = 2,000,000$
The MTA processing mails from the outside world	$10 \times 40,000 = 400,000$
All other directory lookups (like IMAP clients validating certain addresses, and so on)	800,000

To summarize: The total number of directory lookups everyday would be about 3,200,000 (3.2 million). If these lookups were spread out uniformly throughout 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. Table 20–5 shows the shift load for the morning, afternoon, and evening periods of an eight hour day.

Table 20–5 Working Hour Loads

Shift Load	Lookups
Morning load	65%: $0.90 \times 0.65 \times 3,200,000 = 1,872,000$ lookups for 2 hours (936,000 lookups every hour)
Afternoon load	10%: $0.90 \times 0.10 \times 3,200,000 = 288,000$ lookups for 1 hour (288,000 lookups every hour)
Evening load	20%: $0.90 \times 0.20 \times 3,200,000 = 576,000$ lookups for 2 hours (288,000 lookups every hour)

These calculations indicate that the 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 look into individual components of the installation and estimate good values for each.

I/O Subsystem Requirements

This section contains these topics:

- [About the I/O Subsystem](#)
- [Rough Estimates of Disk Space Requirements](#)
- [Detailed Calculations of Disk Space Requirements](#)

About the I/O Subsystem

The I/O subsystem can be compared to a pump that sends 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 [Table 20–6](#) to derive a rough estimate of the overall disk requirement.

Table 20–6 *Disk Space Requirements*

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

The data shown in [Table 20–6](#) 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 Oracle Database, the sizing for disk space is primarily a sizing of the underlying database. Oracle Internet Directory stores its data in the tablespaces described in [Table 20-7](#).

Table 20-7 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 Oracle 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 referred to in [Table 20-7](#). All of the size calculations are based on the variables in [Table 20-8](#).

Table 20-8 Variables Used for Size Calculation

Variable Name	Description
<i>num_entries</i>	Total number of entries in the directory
<i>attrs_per_entry</i>	Average number of attributes for each directory entry
<i>avg_attr_size</i>	Average size of the attribute value in bytes
<i>avg_dn_size</i>	Average size of the DN of an attribute in bytes
<i>objectclass_per_entry</i>	Average number of object classes that an entry belongs to
<i>objectclass_size</i>	Average size of the name of each objectclass in bytes
<i>num_cataloged_attrs</i>	Number of cataloged attributes used in the entries

Table 20–8 (Cont.) Variables Used for Size Calculation

Variable Name	Description
<i>entries_per_catalog</i>	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.
<i>change_log_capacity</i>	Number of changes that we wish to buffer for replication purposes
<i>num_acis</i>	Overall number of ACIs in the directory
<i>num_auditlog_entries</i>	Number of auditlog entries to store in the directory
<i>db_storage_ovhd</i>	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.
<i>db_index_ovhd</i>	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.
<i>factor_of_safety</i>	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.
<i>initial_num_entries</i>	Total number of entries that are initially bulk-loaded into the directory
<i>avg_attrname_len</i>	Average size of attribute name, in bytes
<i>num_stats_entries</i>	Number of statistics entries generated by OID Server Manageability when the host DSF attribute 'orclstatsflag' is enables
<i>attrs_per_stats_entry</i>	Average number of attributes for each statistics entry

Using the variables shown in [Table 20–8](#), the size of individual tablespaces can be calculated as shown in [Table 20–9](#).

Table 20–9 Size of Individual Tablespaces

Tablespaces Containing Tables	Formula
ATTRSTORE_INDEX_SIZE	$\text{num_entries} * (\text{attrs_per_entry} + 6) * 10$
CATALOG_INDEX_SIZE	$\text{entries_per_catalog} * \text{num_cataloged_attrs} * \text{avg_attr_size} * \text{db_index_ovhd} + \text{num_entries} * \text{objectclass_per_entry} * \text{objectclass_size} * \text{db_index_ovhd} + \text{num_acis} * 1.5 * \text{avg_dn_size} * \text{db_index_ovhd} + \text{num_auditlog_entries} * 2 * \text{avg_dn_size} * \text{db_index_ovhd}$
CN_SIZE	$\text{num_entries} * \text{avg_dn_size} * \text{db_storage_ovhd}$
DN_INDEX_SIZE	$\text{num_entries} * 2 * (\text{avg_dn_size} * 3)$
DN_SIZE	$\text{num_entries} * 2 * (\text{avg_dn_size} + 4)$
OBJECTCLASSES_SIZE	$\text{num_entries} * \text{objectclass_per_entry} * \text{objectclass_size} * \text{db_storage_ovhd} + \text{num_auditlog_entries} * 2 * \text{avg_dn_size} * \text{db_storage_ovhd}$
OLTS_ATTR_STORE	$(\text{num_entries} * (((\text{attrs_per_entry}) * (\text{avg_attrname_len} + \text{avg_attr_size} + 22)) + 6 * 35) * \text{db_storage_ovhd}) + \text{attrstore_index_size}$
OLTS_BATTRSTORE	$6M + (((\text{num_binary_attrs} * \text{avg_binval_length}) + 6 * 35) * \text{db_storage_ovhd})$
OLTS_CT_STORE	$(\text{cn_size} + \text{objectclasses_size} + \text{dn_size} + \text{catalog_index_size} + \text{dn_index_size})$

Table 20–9 (Cont.) Size of Individual Tablespaces

Tablespaces Containing Tables	Formula
OLTS_DEFAULT	$(\text{change_log_capacity} * 4 * \text{avg_attr_size} * \text{db_storage_ovhd} * \text{db_index_ovhd}) + (\text{initial_num_entries} * 2 * (\text{avg_dn_size} + 4))$
OLTS_SVRMGSTORE	$2\text{M} + \text{num_stats_entries} * ((\text{avg_attrname_len} + \text{avg_attr_size} + 20) * (2 * \text{attrs_per_stats_entry}) * \text{db_storage_ovhd} * (\text{orclstatsperiodicity} / 10) * 12)$
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. [Table 20–10](#) illustrates the values of each variable introduced in this section for Acme Corporation.

Table 20–10 Values for Variables Used for Sizing Calculations

Variable Name	Value
<i>num_entries</i>	1,000,000
<i>attrs_per_entry</i>	20
<i>avg_attr_size</i>	32 bytes
<i>avg_dn_size</i>	40 bytes
<i>objectclass_per_entry</i>	5 (each entry belongs to an average of 5 object classes)
<i>objectclass_size</i>	10 bytes
<i>num_cataloged_attrs</i>	10
<i>entries_per_catalog</i>	1,000,000
<i>change_log_capacity</i>	80,000 changes (2 for each user)
<i>num_acis</i>	80,000 ACIs (2 for each user)
<i>num_auditlog_entries</i>	1000
<i>db_storage_ovhd</i>	1.4 (40% overhead)
<i>db_index_ovhd</i>	5.0 (400% overhead)
<i>factor_of_safety</i>	1.5 (50% factor of safety)
<i>initial_num_entries</i>	1,000,000
<i>num_stats_entries</i>	5
<i>attrs_per_stats_entry</i>	12
<i>orclstatsperiodicity</i>	60 (root DSE attribute)
<i>avg_attrname_len</i>	6

If we now plug these values into the equations described earlier, we get the values listed in [Table 20–11](#).

Table 20–11 *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
OLTS_SVRMGSTORE	2,701,568	3
SYSTEM	314572800	300
Total Size	5038093862	4920

[Table 20–11](#) 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:

$$N * [150 + \{ \text{attrs_per_entry} + 6 \} * (\text{avg_attrname_len} + \text{avg_attr_size} + 40)] * 1.3$$

See Also: [Chapter 21, "Tuning Considerations for the Directory"](#) for further information on SGA tuning

[Table 20–12](#) gives minimum memory requirements for different directory configurations.

Table 20–12 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). [Table 20–13](#) 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 20–13 Maximum Possible Throughput for Two Types of Operations

Percent Available Bandwidth	Operations/sec		Operations/sec	
	1024 bytes		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:

- [CPU Configuration](#)
- [Rough Estimates of CPU Requirements](#)
- [Detailed Calculations of CPU Requirements](#)

CPU Configuration

The CPU sizing for Oracle Internet Directory is directly a function of the user workload. The following factors will determine CPU configuration:

- The number of concurrent operations you want to support. This will be directly dependent on the number of users performing operations simultaneously.
- The acceptable latency of each operation. For example, in an 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 typically co-resides with the Oracle Database, we recommend at least a two-CPU system. We give the rough estimates in [Table 20-14](#) based on the level of usage of Oracle Internet Directory.

Table 20–14 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 10g Release 2 (10.1.2):

$\text{SPECint_rate95 baseline} = 0.5 * (\text{max \# of concurrent operations at peak throughput})$

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 10g Release 2 (10.1.2):

$\text{SPECint_rate95 baseline} = 0.4 * (\text{throughput of operations at max supported concurrency})$

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.

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](#)
- [Tools for Performance Tuning](#)
- [CPU Usage Tuning](#)
- [Memory Tuning](#)
- [Disk Tuning](#)
- [Database Tuning](#)
- [Entry Caching](#)
- [Optimizing Searches](#)
- [Setting the Time Limit Mode](#)
- [Setting the Timeout for Client/Server Connections](#)

See Also: ["Troubleshooting Directory Performance"](#) on page J-9

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 Microsoft Windows:

Tool	Description
Windows Performance Monitor	Provides a customized view of the events in the system
Windows Task Manager	Provides a high level output (like <code>top</code> on UNIX) of the major things happening in the system.

Knowledge of the following tools is recommended for the Oracle Database:

- `utlbstat.sql` and `utlestat.sql`, or `statspack`
- The `ANALYZE` function in the `DBMS_STATS` package

See Also:

- *Oracle Database Reference* in the Oracle Database Documentation Library for information about `utlbstat.sql` and `utlestat.sql`
- *Oracle Database Performance Tuning Guide* for information about `statspack`
- *Oracle 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.sql`), 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 8.6 or later. Visit: <http://www.datafocus.com/>
-

See Also: The "oidstats.sql" command-line tool reference in *Oracle Identity Management User Reference*

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 under utilized, there is a significant amount of idle time in the system, and this idle time cannot be eliminated at even higher loads.

Internal benchmarks show that Oracle Internet Directory performs best when approximately 70 to 75 percent of the CPU resources are consumed by Oracle Internet Directory processes, and the remaining (about 25 to 30 percent) are consumed by the Oracle foreground processes corresponding to the database connections. While monitoring CPU usage, it is also important to monitor the percentage of time spent in the system space compared to user space. Internal benchmarks show best throughput numbers at about 85 percent user and 15 percent system time.

This section contains these topics:

- [Tuning CPU for Oracle Internet Directory Processes](#)
- [Tuning CPU for Oracle Foreground Processes](#)
- [Taking Advantage of Processor Affinity on SMP Systems](#)
- [Other Alternatives for a CPU Constrained System](#)

Tuning CPU for Oracle Internet Directory Processes

The demands placed by Oracle Internet Directory processes on the CPU can be controlled by the `ORCLSERVERPROCS` and `ORCLMAXCC` parameters. [Table 21-1](#) lists suggested values for these parameters for various client loads.

Table 21-1 Suggested Values for ORCLSERVERPROCS and ORCLMAXCC Parameters

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 sixteen database connections (10+5+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 throughput 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.sql` 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 8.6 or later. 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 Oracle Internet Directory 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 Oracle 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 Oracle 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 the Oracle Database](#)
- [Other Alternatives for a Memory-Constrained System](#)

Tuning the System Global Area (SGA) for the Oracle Database

The SGA should be sized based on the available physical memory on the system running the Oracle Database.

See Also: *Oracle 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: *Oracle 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.

[Table 21-2](#) gives a quick overview of the recommended values of RDBMS parameters for various client loads. These parameters are configurable in the initialization parameter file.

Table 21–2 Recommended RDBMS Values for Various Client Loads

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 Parameters](#)
- [Parameters Dependent on Oracle Internet Directory Server Configuration](#)
- [SGA Parameters Dependent on Hardware Resources](#)

Required Parameters

Configure the OPEN_CURSORS parameter as follows:

```
OPEN_CURSORS=200
```

The Oracle Database 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.

Oracle Internet Directory replication depends on database replication. If you are using Oracle Internet Directory replication, configure the JOB_QUEUE_PROCESSES and PARALLEL_MIN_SERVERS database parameter to 1 or greater. For example:

```
JOB_QUEUE_PROCESSES=1
PARALLEL_MIN_SERVERS=1
```

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 *Oracle Database Net Services Administrator's Guide* and *Oracle 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 21-5. 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, 10g Release 2 (10.1.2), 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 doubles 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, using the multiprocess directory server instance and the Oracle buffer cache results in greater scalability.

See Also: ["Setting System Operational Attributes"](#) on page 5-7 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 several thousand attribute 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. If you do not need to retrieve all the attributes of the group entry, specify `required attributes` in the search request to optimize the latency.
2. If you still see unacceptable latency, even with `required attributes` specified, try re-creating the index `EI_ATTRSTORE`.
 - a. Stop the Oracle Internet Directory instance.
 - b. Log in to the Oracle Internet Directory database as the ODS user and execute the following query:

```
DROP INDEX EI_ATTRSTORE;

CREATE BITMAP INDEX EI_ATTRSTORE ON
DS_ATTRSTORE(ENTRYID,ATTRNAME)
TABLESPACE OLTS_ATTRSTORE
PARALLEL COMPUTE STATISTICS NOLOGGING;
```

- c. Start the Oracle Internet Directory instance.

Employ this method carefully and only when required, as re-creating `EI_ATTRSTORE` in this fashion may degrade the performance of typical messaging search operations by 5-10%.

Optimizing Searches for Skewed Attributes

To service a typical search request, the directory server sends a SQL statement to the Oracle Database. 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 A-38](#) on page A-26.
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 5-7, 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. 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**.

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](#)
- [Modifying Oracle Internet Directory Garbage Collectors](#)
- [Enabling and Disabling Logging for Oracle Internet Directory Garbage Collectors](#)

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. The Oracle Internet Directory statistics collector also uses the Oracle Internet Directory garbage collection framework.

This section contains these topics:

- [Components of the Oracle Internet Directory Garbage Collection Framework](#)
- [How Oracle Internet Directory Garbage Collection Works](#)
- [Garbage Collector Entries and the Oracle Internet Directory Statistics Collector Entry](#)
- [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 background database processes.

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=subconfigsentry.
- The DN of the plug-in trigger used to modify a garbage collection job is:
cn=Modify PurgeConfig,cn=plugin,cn=subconfigsentry.
- The DN of the plug-in trigger used to delete a garbage collection job is:
cn>Delete PurgeConfig,cn=plugin,cn=subconfigsentry.

See Also: "Oracle Internet Directory Configuration Schema Elements" in *Oracle Identity Management User Reference* for a list and descriptions of the attributes of the garbage collection plug-in

Background Database Processes

The background database processes that are invoked by the garbage collection plug-in include garbage collectors and the Oracle Internet Directory statistics collector.

Garbage Collectors You can set and manage these behaviors of a garbage collector:

- 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=subconfigsentry.
- 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=subconfigsentry.
- 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=subconfigsentry.
- 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=subconfigsentry.
- 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=subconfigsentry.

- 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`.
- 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: "Oracle Internet Directory Configuration Schema Elements" in *Oracle Identity Management User Reference*

Note: Oracle 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.

Oracle Internet Directory Statistics Collector You can set and manage these behaviors of the Oracle Internet Directory statistics collector:

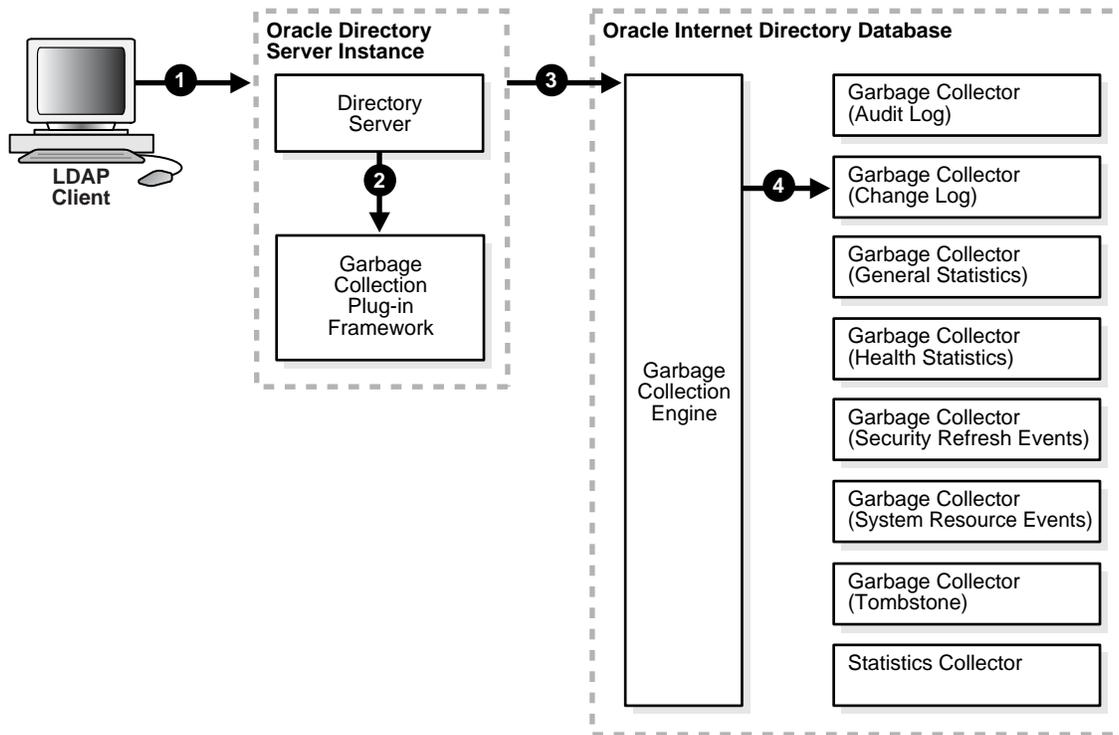
- The time it starts
- How often it runs

The Oracle Internet Directory statistics collector collects statistics about Oracle Internet Directory. The container for this background database process is `cn=oidstats_config, cn=purgeconfig, cn=subconfigsubentry`.

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 22-4 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 background database process—in this case, the change log garbage collector. The background database process runs according to the parameters specified in its configuration set entry.

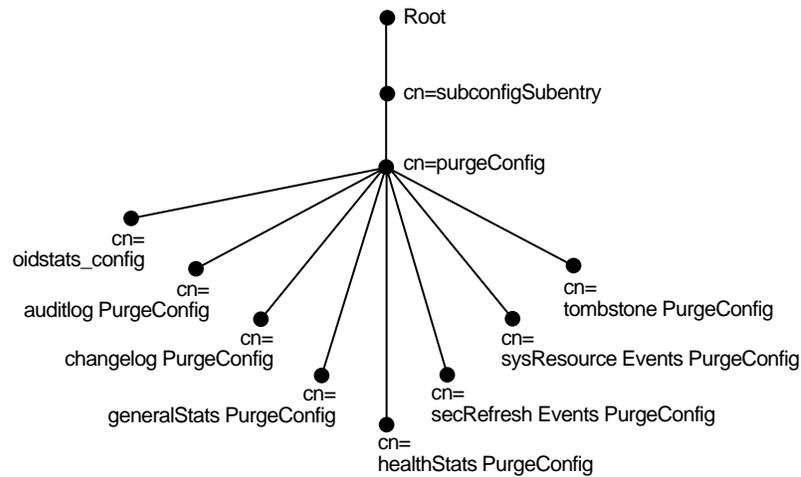
Garbage Collector Entries and the Oracle Internet Directory Statistics Collector Entry

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: "Oracle Internet Directory Configuration Schema Elements" in *Oracle Identity Management User Reference* for a description of each garbage collector attribute

The Oracle Internet Directory statistics collector entry, with its attributes, is also located in the entry `cn=purgeconfig`, immediately below the entry `cn=subconfigsubentry`.

[Figure 22–2](#) shows the location of these entries.

Figure 22–2 Garbage Collection Entries in the DIT

Change Log Purging in Multimaster Replication

Both replication and Oracle Directory Integration and Provisioning use change logs to propagate information from a supplier directory to a consumer directory. All change logs are stored in the table `ods_chg_log`. In addition, replication change logs are stored in `asr_chg_log`. When the change log garbage collector runs, it purges change logs that are no longer needed by any change log consumers. This prevents the change log store in the Oracle Internet Directory database from becoming too large.

Change Log Purging Methods

The change log garbage collector uses the following two methods to determine which change logs to purge:

- Change number-based purging

Change number-based purging respects the change status of all change log consumers. That is, it does not purge change logs unless they have been consumed by all consumers. When the change log garbage collector runs, it purges all change logs that have been consumed by replication, Oracle Directory Integration and Provisioning, and other consumers.

- Time-based purging

Time-based purging is a fall-back method designed to purge change logs of a certain age. It ensures that old change logs are purged even if they have not been consumed by all change log subscribers. Time-based purging respects the change status of replication, but not the change status of other consumers. The change log garbage collector purges all change logs that are not needed by replication and that are at least `orclpurgetargetage` hours old. If `orclpurgetargetage` is zero, the change log garbage collector does this immediately. If `orclpurgetargetage` is an invalid number or not defined, the default value is 240 hours (10 days). Change logs needed by replication are not purged until they have been consumed by replication.

If you have deployed Oracle Directory Integration and Provisioning, and you want to enable time-based purging, be sure to set `orclpurgetargetage` to a large enough value to allow change logs to be processed by Oracle Directory Integration and Provisioning before they are purged. A value of 240 will allow 10 days before change logs are purged.

Configuring Change Log Purging

You configure time-based purging by modifying the `orclpurgetargetage` attribute of the changelog purging configuration entry. This example configures time-based purging for 120 hours (5 days). Use an LDIF file similar to this:

```
dn: cn=changelog purgeconfig,cn=purgeconfig,cn=subconfigsubentry
changetype:modify
replace: orclpurgetargetage
orclpurgetargetage: 120
```

To apply the ldif file `mod.ldif`, type:

```
ldapmodify -p port -h host -D dn -w password -f mod.ldif
```

Note: The container for the change log garbage collector is `cn=changelog purgeconfig, cn=purgeconfig, cn=subconfigsubentry`.

See Also: "Oracle Internet Directory Configuration Schema Elements" in *Oracle Identity Management User Reference*

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 A-9](#) on page A-6.
3. Choose **Apply**.

Modifying a Garbage Collector by Using Command-Line Tools

This section provides examples of how to modify garbage collectors by using command-line tools. The garbage collection attributes that you can modify are listed in "Oracle Internet Directory Configuration Schema Elements" in *Oracle Identity Management User Reference*.

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: orclpurgenow
```

```
orclpurgenow: 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
```

Modifying the Oracle Internet Directory Statistics Collector

You modify the Oracle Internet Directory statistics collector in the same way as the garbage collectors, but there are only three modifiable fields. See "[Oracle Internet Directory Statistics Collection Management Fields in Oracle Directory Manager](#)" on page A-6.

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
- The operation status
- The time stamp
- Any errors caught

To enable logging of garbage collection information:

1. Set the `orclpurgedebg` attribute to 1 if needed. When `orclpurgedebg` is set to 1, extra debugging detail information will be logged. This is useful for troubleshooting garbage collection problems.
2. Set the `orclpurgefilename` attribute to a valid file name for the log file, for example: `oidgc001.log`.

3. Set the `orclpurgefileloc` attribute to the path name of the directory in which the log file is located, for example: `/private/qzhou/oracle/ldap/log`.
4. Enable PL/SQL I/O access to the directory specified in step 3. To do this, include the following in the database:

```
UTL_FILE_DIR=PATH_NAME
```

where `PATH_NAME` is the path you specified in Step 3.

See Also: The section on the `UTL_FILE_DIR` parameter type in the *Oracle Database Reference*

5. Shut down the replication server, then the Oracle Internet Directory server.
6. Restart the database.
7. Start the Oracle Internet Directory server, then the replication server.

Disabling Logging for Oracle Internet Directory Garbage Collectors

To disable logging of garbage collection information, set the `orclpurgedebug` attribute to 0.

Note: Even when `orclpurgedebug` is set to 0, minimal information about garbage collector operation is still logged to indicate the garbage collector's activities.

Monitoring Garbage Collection Logging

The information in the garbage collection log can be useful for monitoring and troubleshooting garbage collection. You determine the location of the log by setting attributes when enabling logging. For example, if you configured:

```
orclpurgefilename = oidgc001.log
orclpurgefileloc   = /private/qzhou/oracle/ldap/log
```

when you enabled logging, then you can monitor change log garbage collection activities by reading the file

```
/private/qzhou/oracle/ldap/log/oidgc001.log.
```

The following is an example of the information logged when an administrator modified the `orclpurgenow` attribute of the change log garbage collection configuration entry:

```
Running Garbage Collector: cn=changelog purgeconfig
Starting time: 2005/03/24 11:03:23
PurgeConfig object located, Eid= 936
purge_ODSChglog: Nothing to be purged(no_work_to_do)
purge_ODSChglog: 107 chglogs successfully purged
purge_ASRChglog: Nothing to be purged(no_work_to_do)
purge_ASRChglog: 0 chglogs successfully purged
purge_ASRChglog: 0 chglogs successfully purged

Modifying Garbage Collector for at "2005/03/24 11:03:23
Garbage Collector DN recognized, rdn=cn=changelog purgeconfig
orclPurgeNow successfully retrieved.
Garbage Collector job found: jobno=21
Garbage Collector has been run
Garbage collector is updated successfully!
```

Modifying `orclpurgenow` forces the change log garbage collector to run immediately. As shown in the first paragraph, 107 change logs were purged from the `ods_chg_log` table and 0 change logs were purged from the `asr_chg_log` table. Also, the information in the second paragraph indicates successful modification of `orclpurgenow` attribute.

Migration of Data from Other Data Repositories

This chapter explains how to migrate data from both LDAP Version 3-compatible directories and application-specific data repositories into Oracle Internet Directory.

This chapter contains these topics:

- [The Default Directory Structure of Oracle Internet Directory](#)
- [Migrating Data from LDAP-Compliant Directories](#)
- [Migrating User Data from Application-Specific Repositories](#)

The Default Directory Structure of Oracle Internet Directory

During an Oracle Internet Directory installation, Oracle Universal Installer creates a default schema and directory information tree (DIT). This default DIT framework is described in [Chapter 2, "Directory Concepts and Architecture"](#) and in [Chapter 19, "Deployment of Oracle Identity Management Realms"](#). The framework is flexible and you can modify it to suit the needs of your deployment.

In Oracle Internet Directory 10g Release 2 (10.1.2), 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.

Migrating Data from LDAP-Compliant Directories

This section provides practical information for migrating data from an LDAP-compliant, third-party directory to Oracle Internet Directory. 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:

- [Tools](#)
- [Common Usage Scenarios](#)
- [Tasks For Migrating Data from LDAP-Compliant Directories](#)

Tools

Three tools are commonly used for migrating third-party LDAP data to Oracle Internet Directory. They are:

- [bulkload](#)
- [dipassistant](#)
- [Oracle Directory Integration and Provisioning Server](#)

Which tool or tools you use depends on several factors, including the size of the data you are importing and whether the data must be mapped.

bulkload

The Bulk Loading Tool, `bulkload`, is a command-line tool for loading a large number of entries into a directory server. It uses Oracle SQL*Loader to load the directory entries. The `bulkload` tool expects the input file to be in LDAP Data Interchange Format (LDIF). The `bulkload` tool can validate LDIF input for referential integrity, but it cannot perform any mapping or other transformation on the data.

[Table 23–1, "Features of bulkload and dipassistant"](#) lists the features of `bulkload`, as compared with `dipassistant bootstrap`. For `bulkload` syntax information and examples, see "Oracle Internet Directory Data Management Tools" in *Oracle Identity Management User Reference*.

dipassistant

The Directory Integration and Provisioning Assistant, `dipassistant`, is a command-line tool for administering the synchronization profiles scheduled by the Oracle directory integration and provisioning server. An administrator can use the `dipassistant bootstrap` operation to perform the initial migration of data between a connected directory and Oracle Internet Directory when configuring the Oracle directory integration and provisioning server to perform ongoing synchronization. You also use it for a one-time data migration, without ongoing synchronization.

The `dipassistant bootstrap` operation can take data either directly from a third-party LDAP-compliant directory or from an LDIF file, tagged file, or CSV file. You must provide mapping rules, either as a synchronization profile or in a configuration file.

[Table 23–1, "Features of bulkload and dipassistant"](#) lists the features of `dipassistant bootstrap`, as compared with `bulkload`. For `dipassistant`

syntax information, configuration file properties, information about input file types, and examples, see "Oracle Directory Integration and Provisioning

Tools" in *Oracle Identity Management User Reference* and *Oracle Identity Management Integration Guide*.

Table 23–1 Features of bulkload and dipassistant

Feature	bulkload	dipassistant
Speed	Fast	Slow
Data transfer method	SQL	LDAP
Input types accepted	LDIF file only	LDIF file, LDAP directory, tagged file, CSV file
Transforms data	No	Yes
Validates LDIF input	Yes	No

Oracle Directory Integration and Provisioning Server

Under some circumstances, an administrator might choose not to use dipassistant when configuring the Oracle directory integration and provisioning server. Once configured, the Oracle directory integration and provisioning server itself can migrate data from a connected directory to Oracle Internet Directory. You can also use the Oracle directory integration and provisioning server for a one-time data migration. For more information, see *Oracle Identity Management Integration Guide*.

Common Usage Scenarios

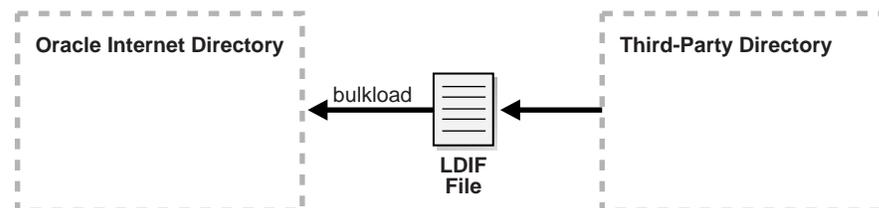
This section describes different scenarios for using the tools described in the "Tools" section. They include:

- [Scenario 1: Using an LDIF File and bulkload](#)
- [Scenario 2: Using dipassistant Directly](#)
- [Scenario 3: Using an LDIF File and dipassistant](#)
- [Scenario 4: Using dipassistant, bulkload, and LDIF Files](#)
- [Scenario 5: Using the Oracle Directory Integration and Provisioning Server](#)

Scenario 1: Using an LDIF File and bulkload

When no translation is required and data is very large (500,000 or more), bulkload is the best choice for migrating data from a third-party directory to Oracle Internet Directory. It is fast and it can validate LDIF input. To use this method, you must first export data from the third-party directory to an LDIF file, as shown in [Figure 23–1, "Using an LDIF File and bulkload"](#).

Figure 23–1 Using an LDIF File and bulkload



LDIF is the IETF-sanctioned ASCII interchange format for representing LDAP-compliant directory data as a file. All LDAP-compliant directories should have tools to export their contents into one or more LDIF files representing the DIT at the time of export.

See Also: RFC 2849 of the IETF, available for download at:
<http://www.ietf.org>

When using an LDIF file and bulkload to migrate data to Oracle Internet Directory, you must perform the following tasks.

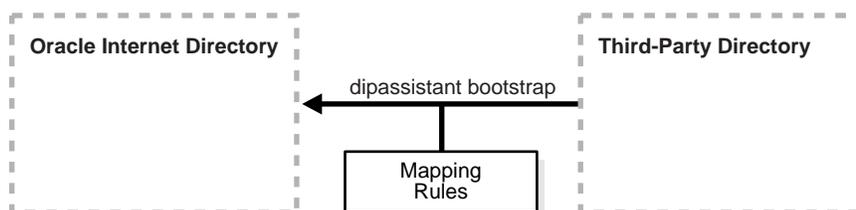
- [Task 1: Export Data from the Non-Oracle Internet Directory Server into LDIF File Format](#)
- [Task 2: Analyze the LDIF User Data for Any Required Schema Additions Referenced in the LDIF Data](#)
- [Task 3: Extend the Schema in Oracle Internet Directory](#)
- [Task 4: Remove Any Proprietary Directory Data from the LDIF File](#)
- [Task 5: Remove Operational Attributes from the LDIF File](#)
- [Task 6: Remove Incompatible userPassword Attribute Values from the LDIF File](#)
- [Task 7: Run the bulkload.sh -check Mode and Determine Any Remaining Schema Violations or Duplication Errors](#)

These tasks are explained in "[Tasks For Migrating Data from LDAP-Compliant Directories](#)" on page 23-6.

Scenario 2: Using dipassistant Directly

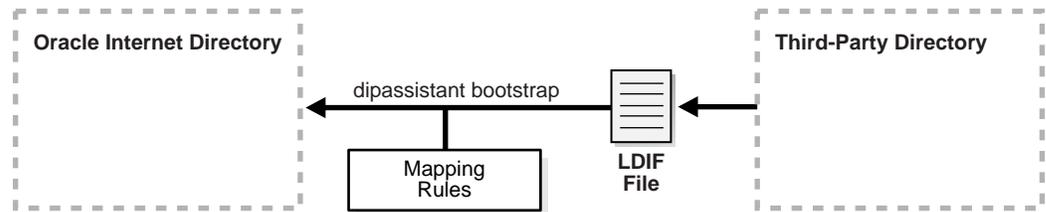
If you must perform mapping when migrating the data from the third-party directory to Oracle Internet Directory, and if the data is small in size, you can use dipassistant. As shown in [Figure 23-2, "Using dipassistant Directly"](#), you can use the third-party directory itself as input to dipassistant.

Figure 23-2 Using dipassistant Directly



Scenario 3: Using an LDIF File and dipassistant

Scenario 3 is a variation on Scenario 2. If you do not have direct access to the third-party directory, you can have the administrator export the data to an LDIF file. As shown in [Figure 23-3, "Using an LDIF File and dipassistant"](#), dipassistant can take its input from an LDIF file. You could also use Oracle directory integration and provisioning server to migrate the data.

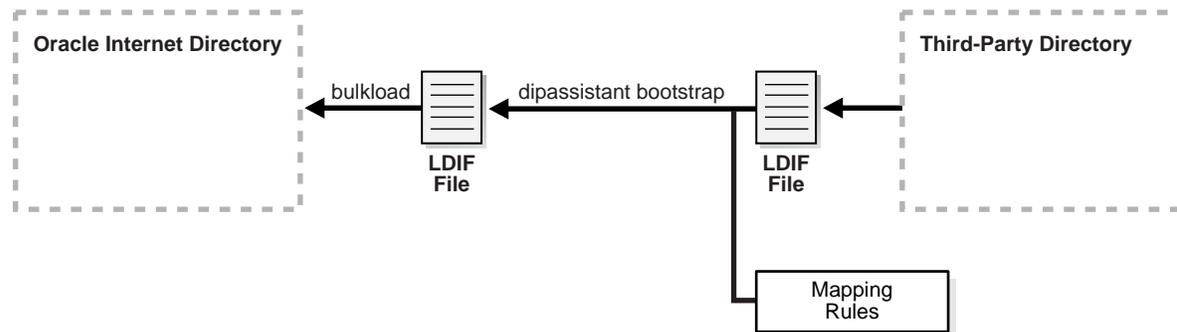
Figure 23–3 Using an LDIF File and dipassistant

Whenever you use an LDIF file and bulkload to migrate data to Oracle Internet Directory, you must perform certain tasks. In this scenario, you will be using a mapping file with `dipassistant` or Oracle directory integration and provisioning, so do not have to perform all the tasks listed in "Scenario 1: Using an LDIF File and bulkload". You only have to perform the following tasks:

- Task 1: Export Data from the Non-Oracle Internet Directory Server into LDIF File Format
- Task 2: Analyze the LDIF User Data for Any Required Schema Additions Referenced in the LDIF Data
- Task 3: Extend the Schema in Oracle Internet Directory

Scenario 4: Using `dipassistant`, bulkload, and LDIF Files

If you have a large amount of data and you must perform mapping on the data, you can use a combination of tools. As shown in Figure 23–4, "Using `dipassistant`, bulkload, and LDIF Files", you can export the data from the third-party directory to an LDIF file, then use `dipassistant bootstrap` to perform the mapping into another LDIF file, which you then load with `bulkload`.

Figure 23–4 Using `dipassistant`, bulkload, and LDIF Files

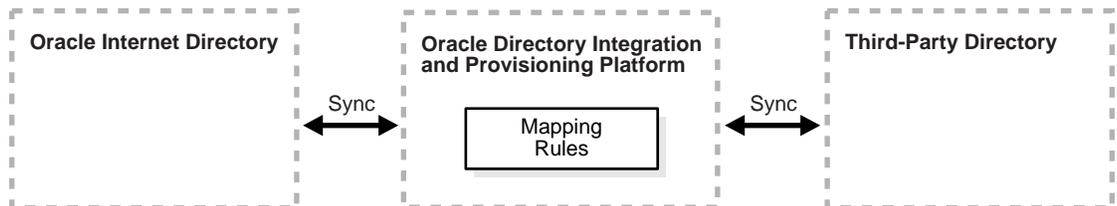
As in Scenario 3: Using an LDIF File and `dipassistant`, you only have to perform these tasks:

- Task 1: Export Data from the Non-Oracle Internet Directory Server into LDIF File Format
- Task 2: Analyze the LDIF User Data for Any Required Schema Additions Referenced in the LDIF Data
- Task 3: Extend the Schema in Oracle Internet Directory

Scenario 5: Using the Oracle Directory Integration and Provisioning Server

The Oracle directory integration and provisioning server allows you to configure bidirectional, ongoing integration between Oracle Internet Directory and a Third-party directory, as shown in [Figure 23-5, "Using the Oracle Directory Integration and Provisioning Server"](#). For more information, see *Oracle Identity Management Integration Guide*.

Figure 23-5 Using the Oracle Directory Integration and Provisioning Server



Tasks For Migrating Data from LDAP-Compliant Directories

To migrate data from LDAP-compliant directories using an LDIF file, you perform the tasks explained in this section.

As described in ["Common Usage Scenarios"](#) on page 23-3, you do not need to perform Tasks 4-7 if you're using `dipassistant` or Oracle directory integration and provisioning with a mapping file.

- [Task 1: Export Data from the Non-Oracle Internet Directory Server into LDIF File Format](#)
- [Task 2: Analyze the LDIF User Data for Any Required Schema Additions Referenced in the LDIF Data](#)
- [Task 3: Extend the Schema in Oracle Internet Directory](#)
- [Task 4: Remove Any Proprietary Directory Data from the LDIF File](#)
- [Task 5: Remove Operational Attributes from the LDIF File](#)
- [Task 6: Remove Incompatible userPassword Attribute Values from the LDIF File](#)
- [Task 7: Run the `bulkload.sh -check` Mode and Determine Any Remaining Schema Violations or Duplication Errors](#)

Task 1: Export Data from the Non-Oracle Internet Directory Server into LDIF File Format

See the vendor-supplied documentation for instructions. If flags or options exist for exporting data from the foreign directory, be sure to select the method that:

- Produces LDIF output with the least amount of proprietary information included
- Provides maximum conformance to the IETF Request for Comments 2849 of the IETF, available for download at: <http://www.ietf.org>

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 8, "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, which is documented in *Oracle Identity Management User Reference Integration Guide*.

If you have users who will be using other Oracle products, you must create users with object classes `orclUserV2` and its required attributes. If you are integrating with Active Directory, you must create users with object class `orclADUser` and its required attributes. These object classes and their attributes are documented in *Oracle Identity Management User Reference*.

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 10g Release 2 (10.1.2) supports the following userPassword attribute hash algorithms:

- No encryption
- [MD4](#)
- [MD5](#)
- [SHA](#)
- SSHA
- [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 8.6 or later. Visit:
<http://www.datafocus.com/>
-
-

See Also: The "bulkload" command-line tool reference in *Oracle Identity Management User Reference* 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 (Idifmigrator).

See Also: The "Idifmigrator" command-line tool reference in *Oracle Identity Management User Reference* 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.

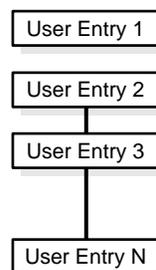
Task 1: Create an Intermediate Template File

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 (Idifmigrator) assigns all of these users to the default identity management realm, which corresponds to the enterprise itself.

Figure 23-6 shows the overall structure of the intermediate template file containing user entries.

Figure 23-6 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 23–2](#) lists and describes the mandatory attributes in a user entry.

Table 23–2 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

Table 23–2 (Cont.) Mandatory Attributes in a User Entry

Attribute	Description
objectclass	Object classes the entry should minimally belong to: <code>inetOrgPerson</code> and <code>orclUserV2</code>

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
- "Object Class Reference for Oracle Identity Management" in *Oracle Identity Management User Reference* for information about optional attributes of the `orclUserV2` object class

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 "Idifmigrator" command-line tool reference in *Oracle Identity Management User Reference* for instructions about using the OID Migration Tool

Part V

Directory Replication

This part provides detailed discussions of replication and high availability and how to plan and manage them. It contains these chapters:

- [Chapter 24, "Oracle Internet Directory Replication Concepts"](#)
- [Chapter 25, "Oracle Internet Directory Replication Administration"](#)

Oracle Internet Directory Replication Concepts

This chapter describes Oracle Internet Directory replication. 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 Oracle Database Advanced Replication Filtering](#)
- [Rules for Partial Replication Filtering](#)

About Directory Replication

This section briefly introduces some of the basic concepts of replication. Subsequent 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. A server that sends the updates made to it to

other servers is known as a supplier. A server that accepts those changes is called a consumer. A server can be both a supplier and a consumer.

A directory replication group can be single-master, multimaster, or fan-out as described in [Table 24-1](#).

Table 24-1 *Types of Directory Replication Groups*

Group	Description
Single-master	Has only one supplier replicating changes to one or more consumers. Only the supplier can be updated, and consumers are read-only.
Multimaster	Also called peer-to-peer or <i>n</i> -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.
Fan-out	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 Oracle Database Advanced Replication or LDAP.

Full and Partial Directory Replication

This section contains these topics:

- [Full Directory Replication](#)
- [Partial Directory Replication](#)

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 Oracle Database Advanced Replication or LDAP.

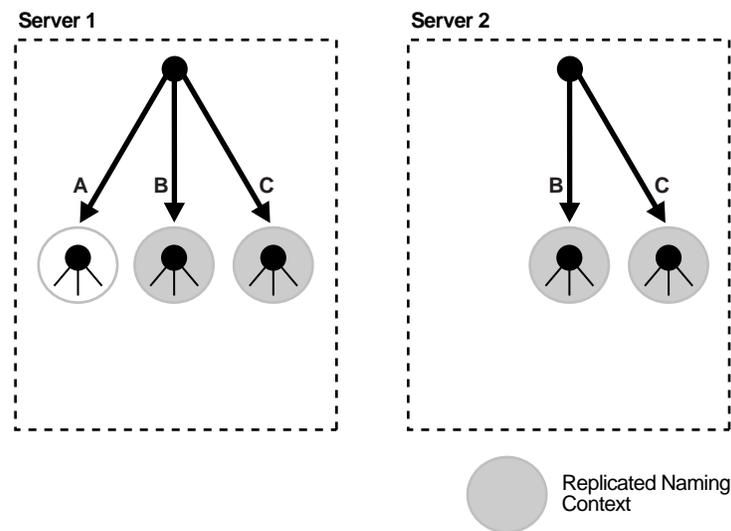
Partial Directory Replication

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 Oracle Database Advanced Replication.

See Also: The "remtool" command-line tool reference in *Oracle Identity Management User Reference*

[Figure 24-1, "Example of Partial Replication"](#) 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, "Example of Partial Replication"](#), 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 24–2](#) compares the two types of replication.

Table 24–2 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
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 24-8

Data Transfer Between Nodes in a Directory Replication Group

In a directory replication group, the protocol for transferring data can be based on either Oracle Database Advanced Replication or LDAP. [Table 24–3](#) shows how each

type handles various features of replication and points to sections of this chapter with more details.

Table 24-3 Types of Data Transfer Between Nodes in a Directory Replication Group

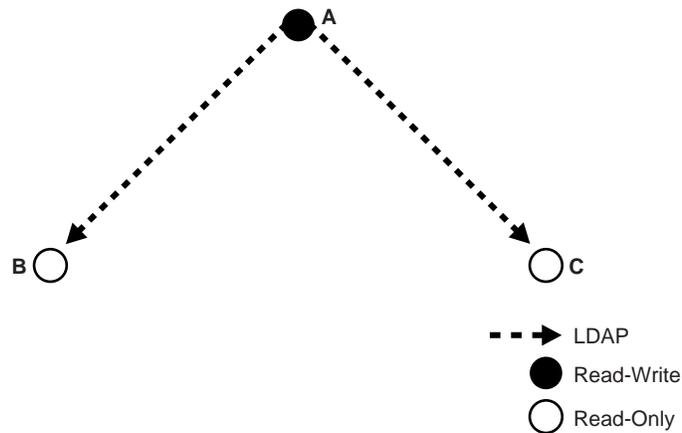
Feature	LDAP-Based Replication	Advanced Replication	More Information
Change propagation	Change propagation from supplier to consumer happens over LDAP	Change propagation from supplier to consumer happens by using Advanced Replication	"Change Logs in Directory Replication" on page 24-15
Replica type supported	Read-only full replica Read-only partial replica	Read/write full replica	"Full Directory Replication" on page 24-2 "Partial Directory Replication" on page 24-2
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 24-4 "Multimaster Replication Groups" on page 24-4 "Fan-Out Replication Groups" on page 24-5

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, "Example of Single-Master Replication" 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.

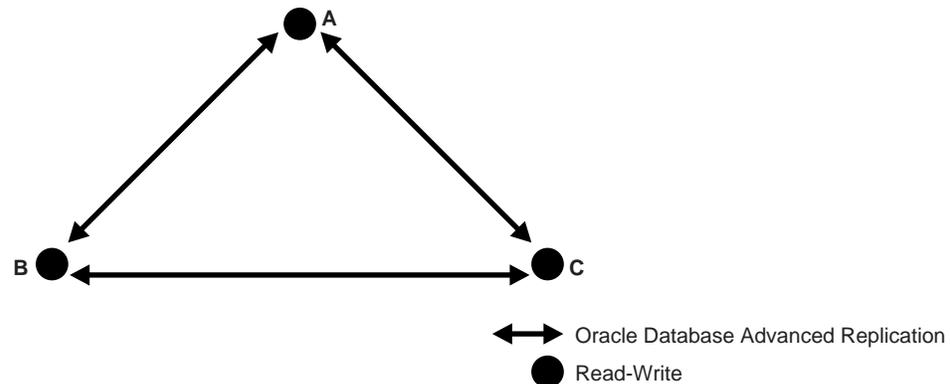
Multimaster Replication Groups

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 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*

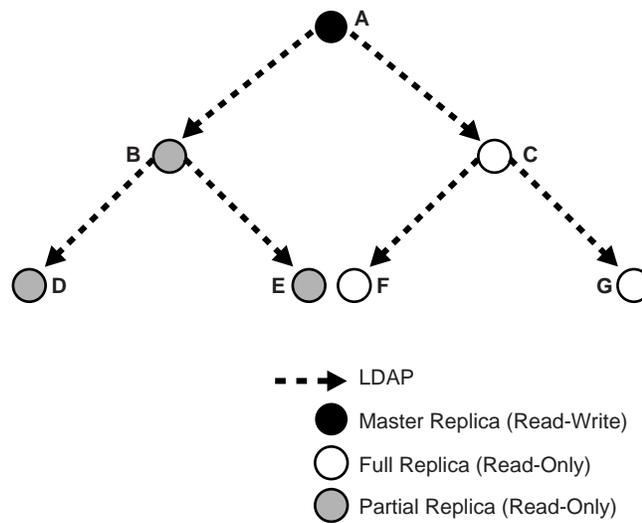
See Also: "[Multimaster Replication](#)" on page 24-16 for more information about multimaster replication

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 24-4 compares multimaster, single-master, and fan-out replication.

Table 24-4 Multimaster, Single-Master, and Fan-Out Replication Compared

Multimaster Replication	Single-Master Replication	Fan-out Replication
Uses only Advanced Replication	Uses LDAP-based replication or Advanced Replication (by switching all masters in a multimaster configuration except one to read-only mode).	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.

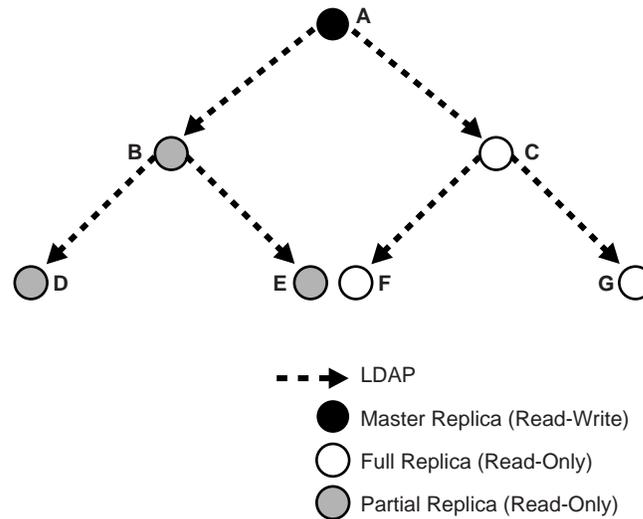
Multimaster Replication with Fan-Out

Beginning with Oracle Internet Directory 10g (9.0.4), Oracle Internet Directory 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 Oracle Database 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 among them by using 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 E is a one-way full replica of Node C. 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 24-21 for more information about fan-out replication

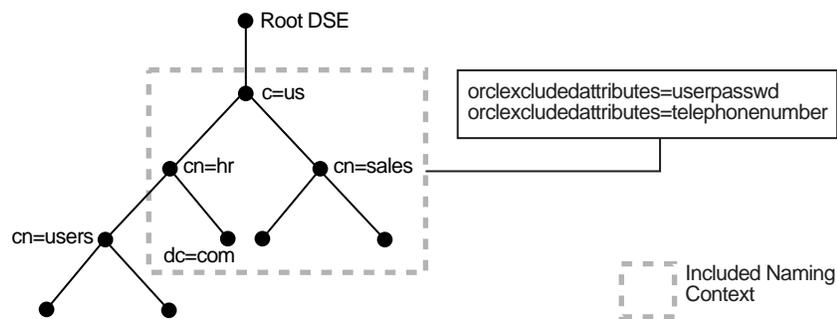
Included and Excluded Naming Contexts

In LDAP-based replication, you can include a given naming context for replication and exclude one or more of the subtrees within that naming context from replication. You can also exclude from replication one or more of the attributes in that naming context.

In Oracle Database Advanced Replication, you can only exclude naming contexts.

In LDAP-based replication, only naming contexts explicitly specified as included are replicated. In Oracle Database Advanced Replication, however, all naming contexts are included by default. To exclude a naming context from replication in Oracle Advanced Replication, specify it using the `orcl_excluded_naming_context` attribute of the Oracle-Advanced-Replication-based-replication agreement entry `orcl_agreement_id=000001`.

Figure 24-6 on page 24-8 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 `c=us`. Within that naming context, one subtree, namely `cn=user1, cn=hr, c=us` is excluded from replication. Moreover, two of the attributes of the `c=us` naming context are excluded from replication—namely, `userPassword` and `telephoneNumber`.

See Also: ["The Replication Naming Context Object Entry"](#) on page 24-11

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 Oracle Database Advanced Replication groups, and those for LDAP-based replication groups.

This section contains these topics:

- [Oracle Database Advanced Replication Agreements](#)
- [LDAP-Based Replication Agreements](#)

Oracle Database Advanced Replication Agreements

For a multimaster replication group, replication agreements are based on 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`.

LDAP-Based Replication Agreements

Unlike multimaster replication groups, in fan-out replication groups there is one replication agreement for each supplier-consumer relationship.

The entry for this kind of replication agreement resides immediately below the replica subentry of 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_agreement,
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_agreement,
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.

Note: The orclreplicadn attribute of an LDAP-based replication agreement specifies the associated consumer node.

See Also: ["The Replication Naming Context Object Entry"](#) on page 24-11

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](#)
- [The Replica Subentry](#)
- [The Replication Agreement Entry](#)
- [The Replication Naming Context Container Entry](#)
- [The Replication Naming Context Object Entry](#)
- [Examples of Replication Configuration Objects in the Directory](#)

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 following is a sample replication configuration container entry:

```
dn: cn=replication configuration
orclaci: access to entry by * (browse)
orclaci: access to attr=(*) by * (search,read)
orclnormdn: cn=replication configuration
cn: replication configuration
description: Replication agreement Container object
```

```
objectclass: top
objectclass: orclcontainerOC
```

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. The local replica entry's `orclreplicaID` matches that of the `orclreplicaID` attribute at the root DSE. For example, in [Figure 24-9](#) on page 24-14, a replica subentry is represented by `orclReplicaID=UID_of_node_D,cn=replication configuration`. The following is a sample replica subentry:

```
dn: orclreplicaid=myhost1_repl1,cn=replication configuration
objectclass: top
objectclass: orclreplicasubentry
orclreplicaid: myhost1_repl1
orclreplicauri: ldap://myhost1:3060/
orclreplicasecondaryuri: ldap://myhost1.mycompany.com:3060/
```

See Also: "Replication Schema Elements" in *Oracle Identity Management User Reference* for descriptions of the attributes of the replica subentry.

The Replication Agreement Entry

This entry contains attributes that define the replication agreement between the two or more nodes and is associated with the `orclReplAgreementEntry` objectclass. There are two kinds of agreement:

1. Oracle Database Advanced Replication Agreement. The replication agreement for Oracle Database Advanced Replication nodes resides under the replication configuration entry. For example: In [Figure 24-8](#), an Oracle Database Advanced Replication agreement entry is represented by `orclagreementID=000001`.
2. LDAP-based replication agreement. The replication agreement for LDAP nodes resides under the supplier's replica subentry. For example, in [Figure 24-9](#), an LDAP-based replication agreement entry is represented by `orclagreementID=000003, orclReplicaID=UID_of_node_D,cn=replication configuration`.

The following is a sample replication agreement entry:

```
dn: orclagreementid=000002,orclreplicaid=myhost1_repl1,
  cn=replication configuration
orclagreementid: 000002
orclupdateschedule: 1
orclreplicationprotocol: ODS_LDAP_1.0
orcllastappliedchangenumber: 0
orclhiqschedule: 10
orclreplicadn: orclreplicaid=myhost2_repl2,cn=replication configuration
orclldapconnkeepalive: 1
objectclass: orclReplAgreementEntry
objectclass: top
```

See Also: "Replication Schema Elements" in *Oracle Identity Management User Reference* 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.

This entry has the RDN `cn=replication namecontext`, and it is created below the `orclagreementID` entry during replication configuration. The following is a sample replication naming context container entry:

```
dn: cn=replication namecontext,orclagreementid=000002,
   orclreplicaid=myhost1_repl1,cn=replication configuration
objectclass: top
objectclass: orclcontainerOC
cn: replication namecontext
```

The Replication Naming Context Object 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 is created below the naming context container entry during replication configuration. It is configurable. For example, in [Figure 24-9](#) on page 24-14, the replication naming context object is `cn=namingcontext001,cn=replication namecontext,orclagreementID=000003,orclReplicaID=UID_of_node_D,cn=replication configuration`

A replication naming context contains these attributes:

- `orclincludednamingcontexts`—The root of the naming context to be replicated
- `orcl'excludednamingcontexts`—Within the included naming context, the root of a subtree to be excluded from replication
- `orcl'excludedattributes`—Within the included naming context, an attribute to be excluded from replication

See Also: "Replication Schema Elements" in *Oracle Identity Management User Reference* for descriptions of the attributes in the replication naming context entry

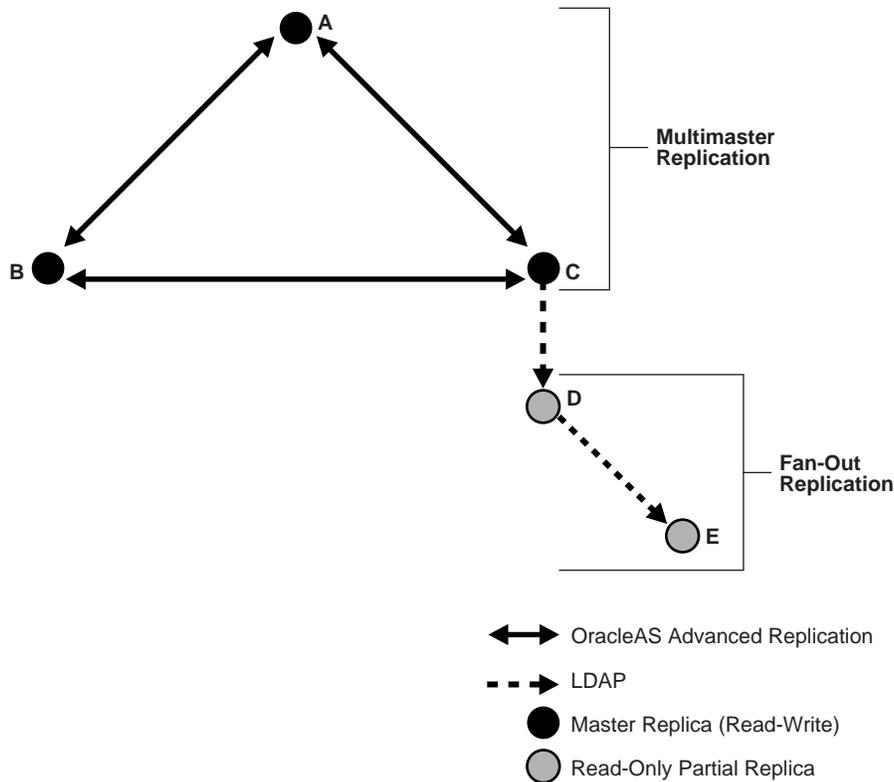
The following is a sample replication naming context object entry.

```
dn: cn=includednamingcontext000001,cn=replication namecontext,
   orclagreementid=000002,orclreplicaid=myhost1_repl1,cn=replication configuration
objectclass: orclreplnamectxconfig
objectclass: top
orclincludednamingcontexts: c=us
orcl'excludednamingcontexts: cn=groups,c=us
orcl'excludedattributes: userPassword
orcl'excludedattributes: telephonenumber
cn: includednamingcontext000001
```

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

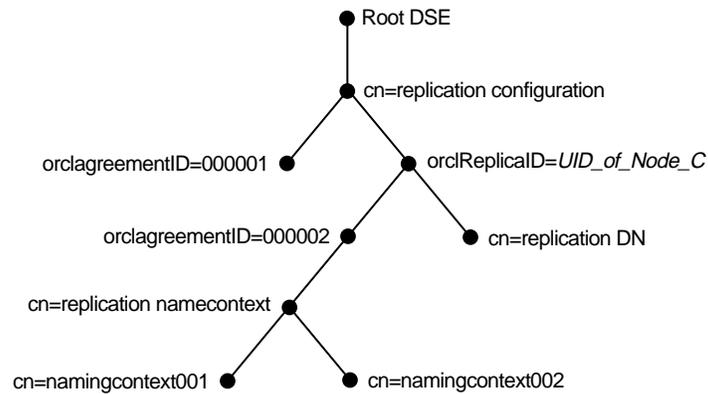


In [Figure 24-7](#), nodes A, B, and C form a multimaster replication group. Node C replicates to a fourth node, D, which, in turn, fans out to Node E.

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 24-12.

Figure 24–8 Example: Replication Configuration Entries for Node C

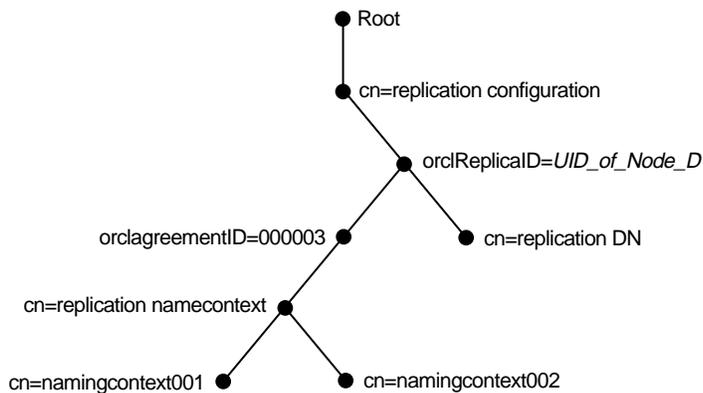
For node C, the entry `cn=replication configuration` at the root DSE contains these RDNs:

- `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`. Its value is the replica entry DN of consumer node D, with which node C has the replication agreement.

- `cn=replication DN`: 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 24-12.

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=replication DN`: 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 `-pchgpwd`, `-presetpwd`, or `-pchgwlpwd` option of the Replication Environment Management Tool. The wallet for replication identity is located at `$ORACLE_HOME/ldap/admin/oidpwrOracle_SID`.

See Also: The "remtool" command-line tool reference in *Oracle Identity Management User Reference*.

Note: In earlier releases, the replication server required the directory server to allow anonymous bind. The replication server no longer requires that.

Secure Sockets Layer (SSL) and Oracle Internet Directory Replication

You can deploy Oracle Internet Directory replication with or without SSL. Replication automatically detects if the target Oracle Internet Directory instance is running at an SSL port. When the replication server binds to the SSL port of an Oracle Internet Directory instance, it will automatically work on top of the Secure Sockets Layer.

Note: In Oracle Internet Directory 10g Release 2 (10.1.2), the Oracle directory replication server cannot communicate directly with an SSL-enabled LDAP server that supports two way (mutual) authentication. The replication server startup will fail and hang if the LDAP server is configured for SSL mutual authentication.

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 Advanced Replication to use SSL encryption, use Oracle Advanced Security.

See Also: *Oracle Advanced Security Administrator's Guide* for information on how to configure 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 `orclastappliedchangenumber` attribute of the replication agreement entry.

- In a replication agreement based on 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.

Change logs are purged by the garbage collector after they have been consumed by the replication server.

See Also: [Chapter 22, "Garbage Collection in Oracle Internet Directory"](#)

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:

- [Oracle Database Advanced Replication](#)
- [Architecture for Multimaster Replication](#)
- [Conflict Resolution in Multimaster Replication](#)

See Also:

["Managing Replication"](#) on page 25-35 for information about how to configure replication agreements

[Appendix F, "The Multimaster Replication Process"](#)

Oracle Database Advanced Replication

In Oracle Database Advanced Replication, the transport of update information between nodes in a replication agreement is managed by Oracle Database Advanced Replication, a store-and-forward transport feature. Advanced Replication enables you to synchronize database tables across two Oracle databases.

Oracle Database 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 Oracle 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 10g Application Server Control Console. 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 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*
- *Oracle Database Advanced Replication* in the Oracle Database Documentation Library for information about Advanced Replication

Architecture for Multimaster Replication

Typical Oracle Database 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.

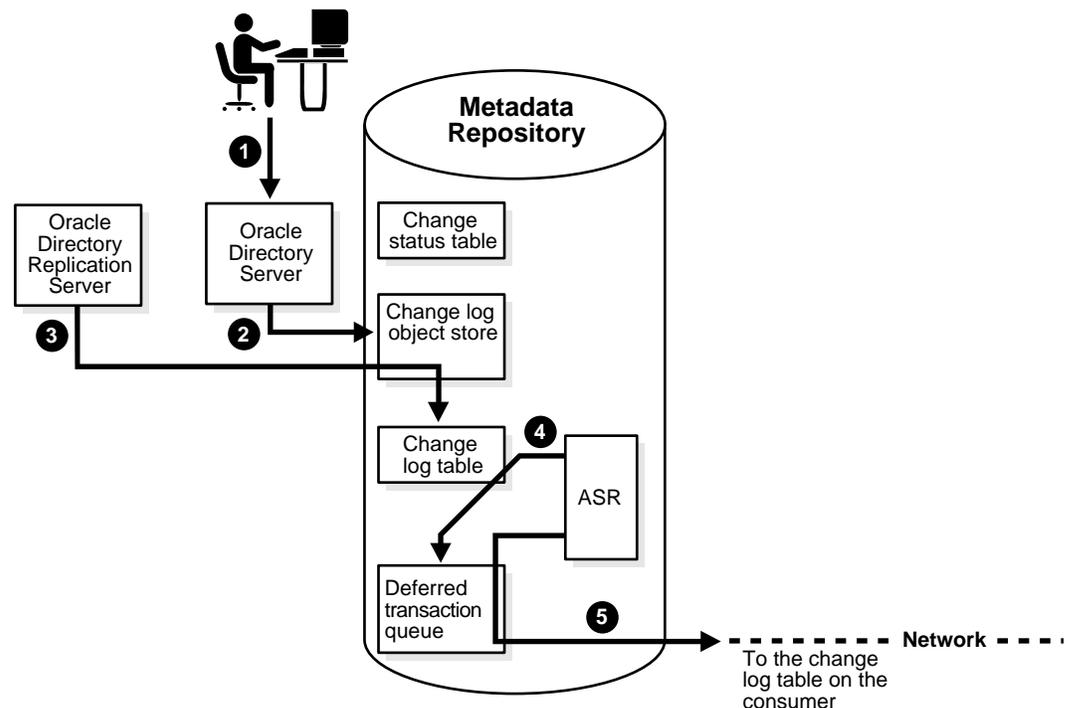
When you configure replication, you specify which nodes in a replication group share changes. Regardless of the number of nodes you introduce into the replication environment, the basic architecture for replication remains the same. Local changes are distributed to a remote master site (RMS), where the 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.

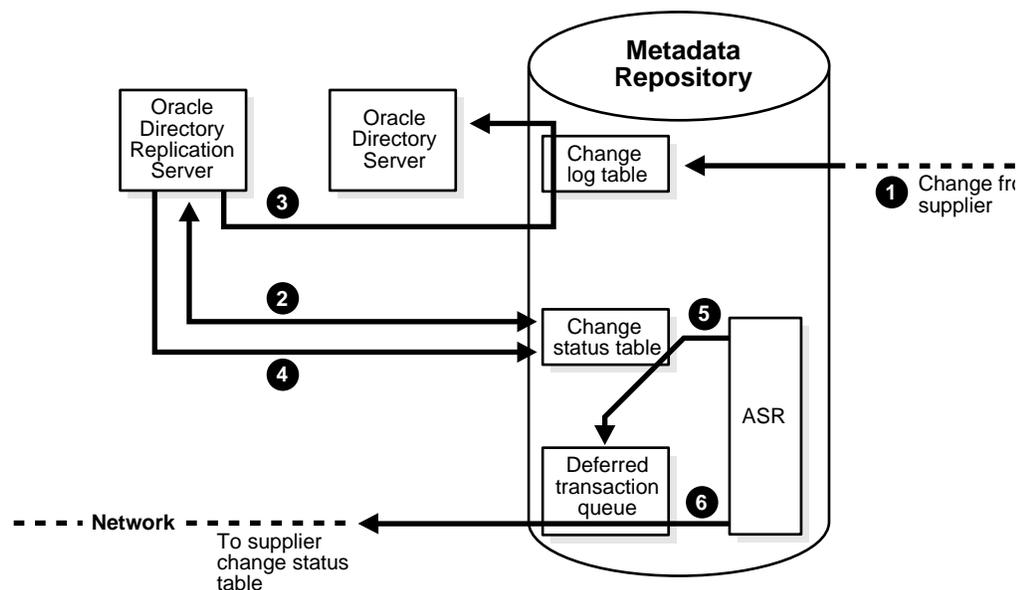
2. The Oracle directory server generates a change log object in the change log object store.
3. At a scheduled time, the Oracle directory replication server launches an outbound change log processing thread. This thread translates the change log object into a row—for example, Change entry—in the change log table.
4. When a change entry is committed to the change log table, Advanced Replication immediately copies the change into the deferred transaction queue.
5. After a scheduled interval, 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 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. Advanced Replication copies the change status update into the deferred transaction queue.
6. After the scheduled Advanced Replication interval, Oracle Database 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 24–5 *Types of Replication Conflict*

Level of Replication Conflict	Description
Entry-level conflicts	<p>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:</p> <ul style="list-style-type: none"> ■ 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 <p>These conflicts can be difficult to resolve. For instance, it may be impossible to resolve a conflict because:</p> <ul style="list-style-type: none"> ■ 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 <p>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.</p>
Attribute-level conflicts	<p>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.</p>

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 multivalued 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 F, "The Multimaster Replication Process"](#) for descriptions of how the multimaster replication process adds, deletes, and modifies entries, and how it modifies DN and RDNs.
- "Oracle Identity Management LDAP Schema Reference" in *Oracle Identity Management User Reference* for schema questions
- The "catalog.sh" command-line tool reference in *Oracle Identity Management User Reference* for catalog questions
- The section on managing group entries in *Oracle Identity Management Guide to Delegated Administration* for group entry questions

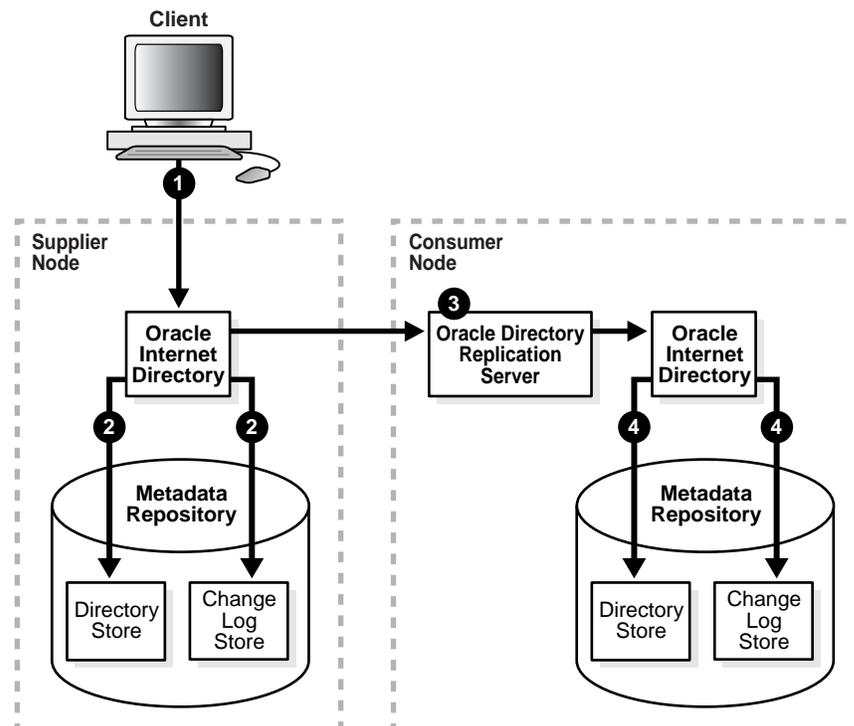
Fan-Out and Partial Replication

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 24-21 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. At the same time, the directory server on the consumer does the following:
 - It makes the required modification, replicating the change in its directory store
 - If the `orclDIPrepository` attribute of its root DSE is TRUE it generates a shadow change log object in its change log store. This attribute is set to TRUE if the objects in this change log store are to be propagated to other fan-out consumers. It is also set to TRUE if the objects are to be propagated to other third party applications by the Oracle Directory Integration Platform server.
 - 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 24-15 for more information about the `orclLastAppliedChangeNumber` attribute

Rules for Oracle Database Advanced Replication Filtering

The following naming contexts cannot be replicated:

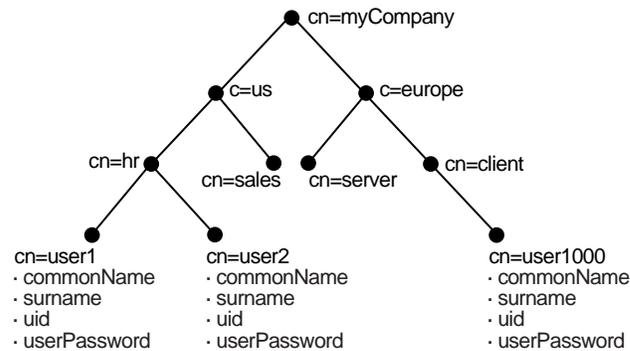
- DSE root-specific entry
- `orclAgreementId=000001,cn=replication configuration`
- `cn=subconfigsentry`
- `cn=Oracle Internet Directory`
- `cn=subregistryentry`

The following naming contexts cannot be excluded from replication:

- `cn=catalogs`
- `cn=subschemasentry`
- `cn=oracleschemaversion`
- `cn=replication configuration`

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](#). A partial list of user attributes is shown under `cn=user1`, `cn=user2`, and `cn=user1000`.

Figure 24–13 A Sample Naming Context

See Also: "Replication Schema Elements" in *Oracle Identity Management User Reference* for descriptions of the attributes in the replication naming context entry

When two or more naming context objects are configured for replication, the filtering rules are as follows:

1. The overall included naming context is the union of all included naming contexts defined in each naming context object.
2. The overall excluded naming contexts is the union of all excluded naming contexts defined in each naming context object.
3. The attribute exclusions in a naming context object are specific only to that naming context object.
4. If there is a conflict between an included naming context and an excluded naming context, the excluded naming context overrules the included naming context. For example, if an included naming context in naming context object A is a subtree of an excluded naming context specified in another naming context object, B, the subtrees specified in `orcl_excluded_naming_contexts` of naming context object B will not be replicated. That is, replication filtering in naming context object A will be ignored.

The following examples show how these rules work:

- [Scenario A: The Included Naming Context in One Naming Context Object Is a Subtree of the Included Naming Context in Another Naming Context Object](#)
- [Scenario B: The Included Naming Context in One Naming Context Object Is a Subtree of An Excluded Naming Context in Another Naming Context Object](#)
- [Rules for Managing Naming Contexts and Attributes](#)
- [Optimization of Partial Replication Naming Context for Better Performance](#)

Scenario A: The Included Naming Context in One Naming Context Object Is a Subtree of the Included Naming Context in Another Naming Context Object

In this scenario, the included naming context in naming context object #2 is a subtree of the included naming context in object #1.

Naming Context Object #1

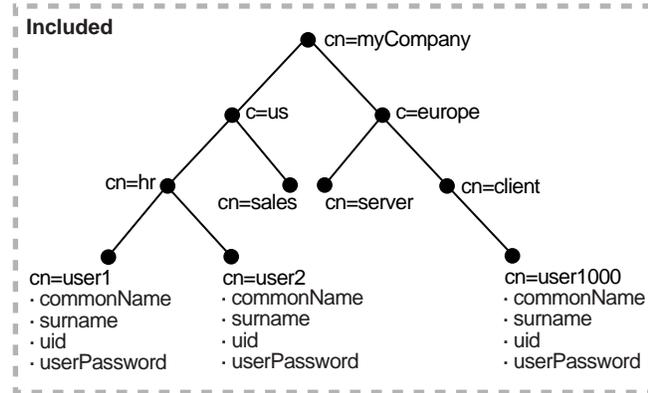
```
dn:cn=namectx001,
cn=replication namecontext,
```

```

orclagreementid=unique_identifier_of_the_replication_agreement,
orclreplicaid=unique_identifier_of_the_supplier,
cn=replication configuration
orclincludednamingcontexts: cn=mycompany
    
```

Naming context object #1 includes the entire DIT under `cn=myCompany`, as shown in [Figure 24-14](#).

Figure 24-14 Naming Context Object #1



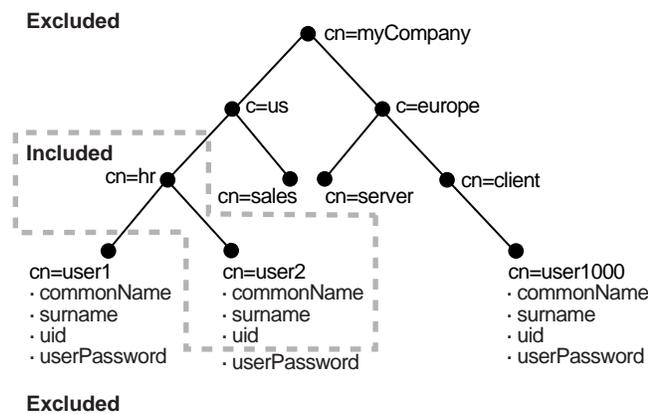
Naming Context Object #2

```

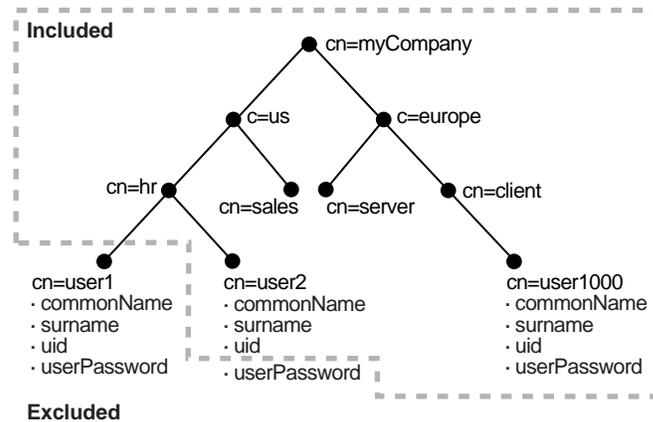
dn:cn=namectx002,
cn=replication namecontext,
orclagreementid=unique_identifier_of_the_replication_agreement,
orclreplicaid=unique_identifier_of_the_supplier,
cn=replication configuration
orclincludednamingcontexts: cn=hr,c=us,cn=mycompany
orclxcludednamingcontexts: cn=user1,cn=hr,c=us,cn=mycompany
orclxcludedattributes: userPassword
    
```

Naming context object #2 includes the DIT under `cn=hr`, `c=us`, `cn=mycompany`, but excludes `cn=user1` and the attribute `userPassword`, as shown in [Figure 24-15](#).

Figure 24-15 Naming Context Object #2



The result of combining naming context objects #1 and #2 is shown in [Figure 24-16](#).

Figure 24–16 Result of Combining Naming Context Objects #1 and #2

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. All changes under the subtree `cn=mycompany` are replicated, except for `cn=user1`, `cn=hr`, `c=us`, `cn=mycompany` and the attribute `userPassword` under `cn=hr`, `c=us`, `cn=mycompany`, which are excluded. The attribute `userPassword` under the rest of the DIT, however, is not excluded from replication because exclusion of `userPassword` was specified only for naming context object #2, which only included the DIT under `cn=hr`.

Scenario B: The Included Naming Context in One Naming Context Object Is a Subtree of An Excluded Naming Context in Another Naming Context Object

In this scenario, 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

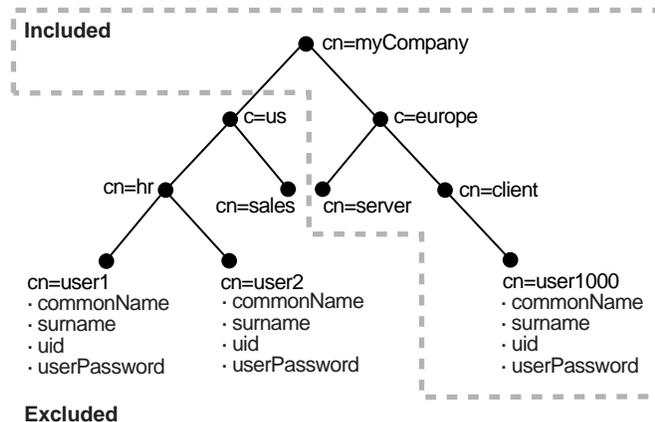
```

dn:cn=namectx001,cn=replication namecontext,
  orclagreementid=identifier,orclreplicaid=supplier,cn=replication configuration
orclincludednamingcontexts: cn=mycompany
orcl'excludednamingcontexts: cn=us,cn=mycompany

```

Naming context object #3 excludes everything under `c=us`, `cn=mycompany`, as shown in [Figure 24–17](#)

Figure 24–17 Naming Context Object #3

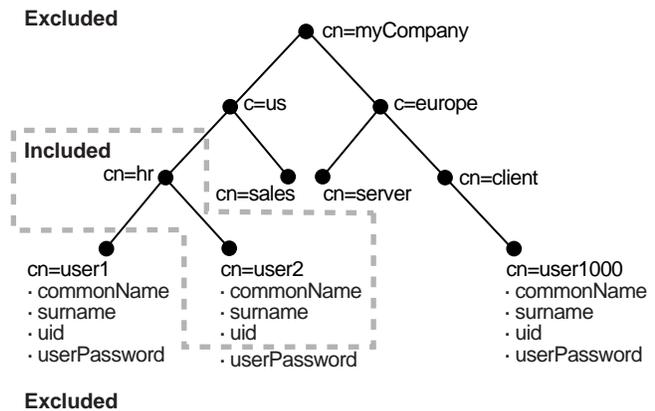


Naming Context Object #4

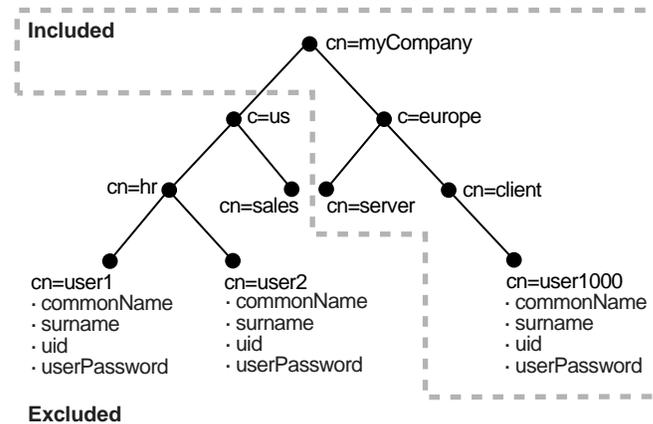
```
dn:cn=namectx002,cn=replication
  namecontext,orclagreementid=identifier,orclreplicaid=supplier,
  cn=replication configuration
orclincludednamingcontexts: cn=hr, c=us,cn=mycompany
orcl'excludednamingcontexts: cn=user1,cn=hr,c=us,cn=mycompany
orcl'excludedattributes: userPassword
```

Naming context object #4 includes the DIT under `cn=hr`, `c=us`, `cn=mycompany` but excludes `user1`, as well as the `userPassword` attribute for all users, as shown in [Figure 24–18](#)

Figure 24–18 Naming Context Object #4



The result of combining naming context objects #3 and #4 is shown in [Figure 24–19](#).

Figure 24–19 Result of Combining Naming Context Objects #3 and #4

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

The following naming contexts cannot be replicated:

- DSE root-specific entry
- `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=oracleschemaversion`
- `cn=replication` configuration

The following attributes cannot be excluded from replication whether they are mandatory or optional. Even if you specify attributes in this list for exclusion from replication, they will always be replicated.

- `orclguid`
- `creatorsname`
- `createtimestamp`
- `cn`
- `dn`
- `attributetypes`
- `objectclasses`
- `objectclass`

- `orclindexedattribute`
- `orclproductversion`

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`.

If you attempt to exclude from replication an attribute that is required for the LDAP operation to be executed, replication will not occur.

If you are configuring partial replication from specific naming contexts in an Oracle Internet Directory node to fan-out replication nodes, then do not change the names of these naming context entries in the source node.

Partial replication will not replicate changes to the root entry of a naming context made by using `ldapmoddn`.

Optimization of Partial Replication Naming Context for Better Performance

You must plan partial replication carefully to avoid degrading the performance of the replication process. For best performance, use as few naming context objects as possible. For example, the combined use of naming context objects #5 and #6 fulfills the same requirement as the use of naming context object #7, but using naming context object #7 provides better performance.

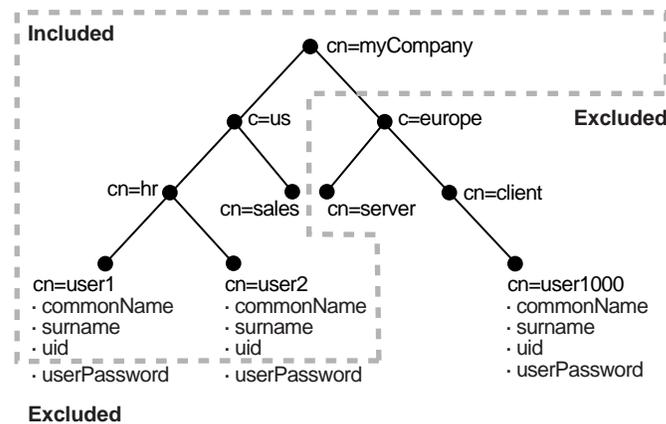
This section contains these examples:

- [Naming Context Object #5](#)
- [Naming Context Object #6](#)
- [Naming Context Object #7](#)

Naming Context Object #5

```
cn=namectx001,cn=replication
namecontext,orclagreementid=identifier,orclreplicaid=supplier,cn=replication
configuration
orclincludednamingcontexts: cn=mycompany
orclexcludednamingcontexts: c=europe,cn=mycompany
orclexcludedattributes: userPassword
```

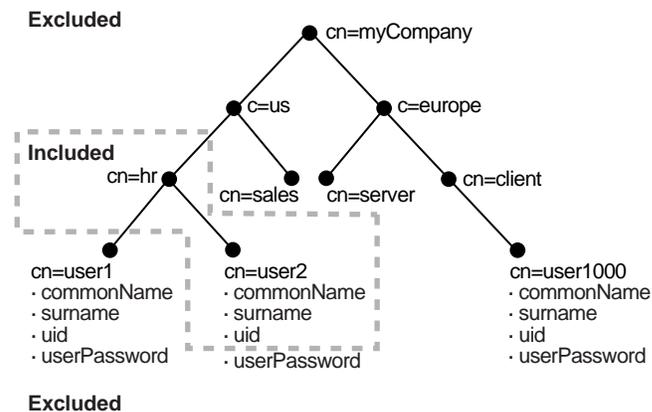
Naming context object #5 is shown in [Figure 24-20](#). It includes the DIT under `cn=mycompany`, but excludes everything under `c=europe`. It also excludes the attribute `userPassword`.

Figure 24–20 Naming Context Object #5**Naming Context Object #6**

```

cn=namectx002,cn=replication
namecontext,orclagreementid=<id>,orclreplicaid=<supplier>,cn=replication
configuration
orclincludednamingcontexts: cn=hr, c=us,cn=mycompany
orclxcludednamingcontexts: cn=user1,cn=hr, c=us,cn=mycompany
orclxcludedattributes: userPassword
  
```

Naming context object #6 is shown in [Figure 24–21](#). It includes the DIT under `cn=hr`, `c=us`, `cn=mycompany` but excludes `user1` and the attribute `userPassword`.

Figure 24–21 Naming Context Object #6

If naming context objects #5 and #6 are combined, then all changes under `cn=mycompany` are replicated, except for `cn=europe`, `c=mycompany`, `cn=user1`, `cn=hr`, `c=us`, `cn=mycompany`, and the attribute `userPassword`.

You could fulfill the same requirement, however, by using naming context object #7. Using a single naming context object provides better partial replication performance.

Naming Context Object #7

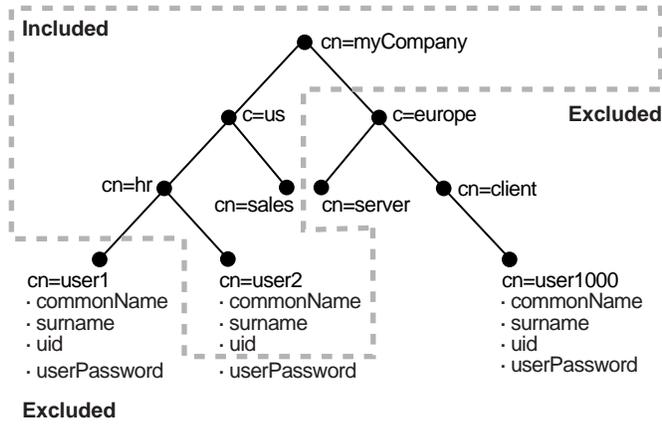
```

cn=namectx001,cn=replication
namecontext,orclagreementid=identifier,orclreplicaid=supplier,cn=replication
configuration
orclincludednamingcontexts: cn=mycompany
orclxcludednamingcontexts: c=europe,cn=mycompany
  
```

orclxexcludednamingcontexts: cn=user1,cn=hr, c=us,cn=mycompany
 orclxexcludedattributes: userPassword

Naming context object #7 is shown in [Figure 24-22](#).

Figure 24-22 Naming Context Object #7



Oracle Internet 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](#)

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:

- [Rules for Configuring Directory Replication Based on Oracle Database Advanced Replication](#)
- [Installing and Configuring a Multimaster Replication Group](#)
- [Adding a Node for Multimaster Replication \(Oracle Database Advanced Replication Types Only\)](#)
- [Deleting a Node from a Multimaster Replication Group](#)
- [Resolving Conflicts Manually in a Multimaster Replication Group](#)

See Also:

- "Installing Oracle Internet Directory in Replicated Mode" in *Oracle Application Server Installation Guide*
- Chapter 11, "Deploying OracleAS Cluster (Identity Management) with Multimaster Replication" in ASHIA

Rules for Configuring Directory Replication Based on Oracle Database Advanced Replication

The following nine rules apply to replication based on Advanced Replication (sometimes referred to as ASR):

1. In this type of Directory Replication Group (DRG), there must be one node identified as the Master Definition Site (MDS): this is the group master. All other nodes taking part in the replication are replicas, which in database replication are termed "Remote Master Sites" (RMS). The MDS must be created as a master node, not as a replica, that is, neither as an Advanced Replication-based replica nor as an LDAP-based replica.

Note: Even though it is not the central master, an Advanced Replication-based replica is sometimes called a remote master site (RMS), due to two facts. The first is that in Advanced Replication, when information is moved from one site to another, the recipient of the transferred information is called a "remote master site." The second fact is that independent changes made directly to an Advanced Replication-based replica are also replicated to all members of its group, making it a "master" during that interaction. Such a group, in which changes to any member are replicated to all other members, is called a multimaster replication group.

See Also:

- Creating a new *master* node is described in ["If you are installing Oracle Internet Directory as a Master"](#) on page 25-4
 - Creating a new *replica* node is described in ["If you are installing Oracle Internet Directory as a Replica"](#) on page 25-5
2. When you install and configure Multimaster replication, the master node for a Directory Replication Group (DRG) and each node that is to become an Advanced Replication-based replica must be initially empty, that is, a new Oracle Internet Directory installation.

Note: If the Master node is not a new installation, use the procedure described in ["Adding a Node for Multimaster Replication \(Oracle Database Advanced Replication Types Only\)"](#) on page 25-13 to add replicas. That procedure will also initialize the replication group.

3. When you add an Oracle Database Advanced Replication-based replica, the new replica must be empty. That is, Oracle Internet Directory must be newly installed.
4. The sponsor node for each Advanced Replication-based replica can be any of the following:
 - A master node
 - An Advanced Replication-based replica of an existing multi-master DRG
 - A supplier of an LDAP replica that is not a consumer LDAP replica of any other LDAP replica
5. An Advanced Replication-based replica cannot be a consumer of an LDAP replica.
6. In Oracle Internet Directory 10g Release 2 (10.1.2), a node cannot be part of more than one multimaster replication group.
7. The data replicated between servers in a directory replication group does not include DSE root-specific data, server configuration data, and replication agreement data.

See Also: ["Rules for Oracle Database Advanced Replication Filtering"](#) on page 24-22

8. When an multimaster replication group is configured, the Oracle Application Server Single Sign-On database schema is automatically configured in replication.
9. You cannot add an Oracle Internet Directory 10g Release 2 (10.1.2) node to an Oracle Internet Directory 10g (9.0.4) DRG. Instead, upgrade all 10g (9.0.4) nodes to 10g Release 2 (10.1.2), then add the new 10g Release 2 (10.1.2) node to the DRG.

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:

[Preliminary Information for Installing and Configuring a Multimaster Replication Group](#)

[Task 1: Install Oracle Internet Directory as a Master on the Master Definition Site \(MDS\) on page 25-6](#)

[Task 2: Install the Oracle Internet Directory as a Replica, on the Remote Master Sites \(RMS\) on page 25-6](#)

[Task 3: Set Up Oracle Database Advanced Replication for a Directory Replication Group on page 25-7](#)

[Task 4 \(Optional\): Load Data into the Directory on page 25-10](#)

[Task 5: Ensure that Oracle Directory Server Instances are Started on All the Nodes on page 25-11](#)

[Task 6: Start the Replication Servers on All Nodes in the DRG on page 25-11](#)

[Task 7: Test Directory Replication on page 25-12](#)

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 for Multimaster Replication \(Oracle Database Advanced Replication Types Only\)"](#) on page 25-13.
 - During entry replication, the directory replication server does not always preserve the spaces between RDN components in the DN. In some rare cases, it may not preserve the case of the letters in the DN.
-
-

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.

Oracle Internet Directory Installation 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.

Later in the installation process, you are prompted to choose an installation type.

- For multimaster replication, you need a single master definition site (MDS). For that, follow the directions in the section entitled "[If you are installing Oracle Internet Directory as a Master](#)" on page 25-4.
- Each other site is a replica, either an Advanced Replication-based replica or an LDAP replica.
 - **Oracle Database Advanced Replication-based replicas** do true multimaster replication: changes made to the master are replicated to the replicas and changes made to any replica are replicated to the master and all other replicas.

Although a site that will be used as an Advanced Replication-based replica can be initially installed as a master, Oracle recommends against that option. Oracle recommends installing each intended replica as a replica right from the start.

Therefore, to create an Advanced Replication-based replica, follow the directions in the section entitled "[If you are installing Oracle Internet Directory as a Replica](#)" on page 25-5, and choose "Advanced Replication" when that choice appears.

- **LDAP replicas** are one-way and read-only: only changes made to the master are replicated to the replica, not from the replica to the master. LDAP replicas can also be used to replicate a portion of the directory information tree rather than the entire directory information tree. In partial replication, you determine what is or is not replicated by defining replica naming context objects.

To create an LDAP replica, follow the directions in the section entitled "[If you are installing Oracle Internet Directory as a Replica](#)" on page 25-5, and choose "LDAP replica" when that choice appears.

See Also: Detailed explanations and examples of setting up and using naming contexts appear in later sections of this chapter.

If you are installing Oracle Internet Directory as a Master

1. Follow the installation procedure documented in the chapter "Installing Oracle Internet Directory in Replicated Mode" in the Oracle Application Server Installation Guide. When the Oracle Universal Installer prompts you to select a product to install, choose the Oracle Application Server Infrastructure.
2. For the Installation Type, select as follows:
 - a. If you do not have (or will not be using) an existing Oracle Application Server Metadata Repository, choose **Identity Management and Oracle Application Server Metadata Repository**.
 - b. If you wish to use an existing Oracle Application Server Metadata Repository, choose **Identity Management**.
3. Ensure that **Oracle Internet Directory** is checked on the **Select Configuration Options** screen.
4. When installing a master, do not check **High Availability and Replication** in the **Select Configuration Options** screen for replication. When you do not check **High Availability and Replication**, Oracle Universal Installer will perform a default Oracle Internet Directory install, that is, it will install a new Oracle Internet Directory as a master node. (You may still check **High Availability and Replication** in the **Select Configuration Options** screen for configuring **Virtual Host** or **OracleAS Clusters** though.)

5. Complete the installation as described in "Installing Oracle Internet Directory in Replicated Mode" in the Oracle Application Server Installation Guide.

If you are installing Oracle Internet Directory as a Replica 1. Follow the installation procedure documented in the chapter "Installing Oracle Internet Directory in Replicated Mode" in the Oracle Application Server Installation Guide. When the Oracle Universal Installer prompts you to select a product to install, choose the Oracle Application Server Infrastructure.

2. For the Installation Type, select as follows:
 - a. If you do not have (or will not be using) an existing Oracle Application Server Metadata Repository, choose **Identity Management and Oracle Application Server Metadata Repository**.
 - b. If you wish to use an existing Oracle Application Server Metadata Repository, choose **Identity Management**.
3. For Configuration Options, ensure that both **Oracle Internet Directory** and **High Availability and Replication** are checked.
4. Because you checked **High Availability and Replication** in the **Select Configuration Options** screen, you will see the **Select High Availability or Replication Option** screen. Select **Replication**.
5. Next, you will see the **Oracle Internet Directory Replication Mode** screen. Choose the type of replica you want to create:
 - For Advanced Replication-based (multimaster) replication, select **Advanced Replication**.
 - For read-only or partial replication, select **LDAP**.
6. On the screen labeled **Oracle Internet Directory Master Node**, specify the hostname and port for the supplier node to be replicated by this node presently being created. If connecting with that node requires SSL protocol, check that box on this screen.
7. Complete the installation as described in Oracle Application Server Installation Guide.

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 DN password
- Modifying the database replication user REPADMIN password
- Displaying various errors and status information for change log propagation

Note: You do not need Advanced Replication to perform partial—that is, LDAP-based—replication.

Even in a directory replication group whose nodes have different patchset versions of Oracle Database 10g, you can replicate if they have the same version of Oracle Internet Directory.

However, if the nodes in a directory replication group are running different versions of Oracle Internet Directory, there is a constraint on modifying directory servers on those nodes: Do not replicate changes generated on a newer version of Oracle Internet Directory to a node that has not yet upgraded to that version. If you do, the changes can contain information that the earlier version cannot properly interpret.

See Also: The "remtool" command-line tool reference in *Oracle Identity Management User Reference* for more information about the Replication Environment Management Tool

Task 1: Install Oracle Internet Directory as a Master on the Master Definition Site (MDS)

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, make sure that each Oracle Internet Directory database instance name is unique on each machine.

To do the actual installation, as a *master*, use the instructions in the section entitled "[If you are installing Oracle Internet Directory as a Master](#)" on page 25-4.

See Also: "Installing Oracle Internet Directory in Replicated Mode" in *Oracle Application Server Installation Guide*.

Task 2: Install the Oracle Internet Directory as a Replica, on the Remote Master Sites (RMS)

Oracle recommends that the sites to be used as Advanced Replications-based replicas be installed initially as replicas, and not as masters.

To do the actual installation, as a *replica*, use the instructions in the section entitled "[If you are installing Oracle Internet Directory as a Replica](#)" on page 25-5.

If an Existing Master is Used as a Remote Master Site Although Oracle recommends starting with empty replicas, you *can* set up replication using machines initially configured as masters rather than replicas. To use a machine initially configured as a master as an RMS, you must first migrate its metadata to the MDS, as follows:

- Make sure the Oracle Internet Directory server is up and running on both the MDS and each such desired replica so that the process (remtool -backupmetadata) can succeed.
- From the newly created node, run the following command:

```
remtool -backupmetadata -master "master_host:master_port/master_repl_dn_pwd" \  
-replica "new_node_host:new_node_port/new_node_repldn_pwd"
```

where *master_host:master_port/master_repdn_pwd* are the hostname, port number, and replication DN password for the desired replica's supplier.

- Apart from loading the metadata into master replica, this tool creates a file named *ocbkup.new_replica_id.TO.master_replicaid.timestamp.dat* containing the metadata as backup. This file is created under the `$ORACLE_HOME/ldap/log` directory. This file contains the changes made to master replica in LDIF format, a copy of SSO container entry [orclApplicationCommonName=ORASSO_SSOSERVER, cn=SSO, cn=Products, cn=OracleContext] and DAS URL container entry [cn=OperationURLs, cn=DAS, cn=Products, cn=OracleContext].

- If the metadata backup succeeded, it displays a message in the terminal:

```
Backup of metadata will be stored in
$ORACLE_HOME/ldap/log/ocbkup.new_replica_id.TO.master_replica_id.timestamp.dat.
Metadata copied successfully.
```

The message will contain the actual path of ORACLE_HOME.

- If an error occurs during this operation, `remtool` reports the error in the terminal from which it was invoked. The error messages are also logged in `$ORACLE_HOME/ldap/log/remtool.log` file.

After successfully migrating the master's metadata to the MDS, you can now safely continue with "[Task 3: Set Up Oracle Database Advanced Replication for a Directory Replication Group](#)" on page 25-7.

Task 3: Set Up Oracle Database Advanced Replication for a Directory Replication Group

The following sections lead you through installing and configuring Advanced Replication by using the Replication Management Tool.

See Also: *Oracle Database Advanced Replication* in the Oracle Database Documentation Library, and the online Help for the Replication Management Tool, for information on configuring Oracle Database Advanced Replication.

To establish a directory replication group (DRG), you must configure the Advanced Replication environment by performing the tasks discussed in these topics:

- [On All Nodes, Prepare the Oracle Net Services Environment for Replication](#)
- [From the MDS, Configure Oracle Database Advanced Replication For Directory Replication](#)

On All Nodes, Prepare the Oracle Net Services Environment for Replication For *each node* in the directory replication group, perform the steps listed here. (Each step is described more fully in the subsections that directly follow this list.)

1. [Configure `sqlnet.ora`.](#)
2. [Configure `tnsnames.ora`.](#)
3. [Stop and restart the listener.](#)
4. [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 = global_database_domain
```

On UNIX, the `sqlnet.ora` file is in `$ORACLE_HOME/network/admin`.

On Microsoft Windows, the `sqlnet.ora` file is in `%ORACLE_HOME%\network\admin`.

2. Configure `tnsnames.ora`.

On each node in the DRG, define all Oracle Internet Directory database instances in the DRG. Each `tnsnames.ora` file must contain connect descriptor information in the following format for each Oracle Internet Directory database:

```
net_service_name =
(DESCRIPTION =
 (ADDRESS =
 (PROTOCOL = TCP)
 (HOST = HOST_NAME_OR_IP_ADDRESS)
 (PORT = port_no_of_listener))
 (CONNECT_DATA =
 (service_name = service_name_of_database)))
```

where `net_service_name` is the global name of the database. For example, if the database global name is `mds.sales.com`, then your `net_service_name` must be `mds.sales.com`. Ensure that your database global name and your `net_service_name` are domain-qualified. In this example, the global name and `net_service_name` are domain-qualified with `sales.com`.

Note:

- The database global name is composed of the `DB_NAME` and `DB_DOMAIN` initialization parameters of your database. For example, if your database's `DB_NAME` is `mds` and `DB_DOMAIN` is `sales.com`, your database global name is `mds.sales.com`. The global name will not be domain qualified if the `DB_DOMAIN` initialization parameter is not defined.
 - The value of the `NAMES.DEFAULT_DOMAIN` parameter in the `sqlnet.ora` file must match the value of the `DB_DOMAIN` initialization parameter of the database.
-
-

See Also: *Oracle Database Net Services Reference* for more information on `tnsnames.ora` syntax.

On UNIX, the `tnsnames.ora` file is in `$ORACLE_HOME/network/admin`.

On Microsoft Windows, the `tnsnames.ora` 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. 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]
```

```
quit
```

4. Test Oracle Net connections to all nodes from each node in the DRG.

IMPORTANT: Try to connect using both of these commands:

```
sqlplus ods/ods_password@net_service_name_without_domain_name
sqlplus ods/ods_password@net_service_name_with_domain_name
```

If you cannot connect, then replication will not work.

From the MDS, Configure Oracle Database Advanced Replication For Directory Replication To do this:

1. From the MDS console, connect as the system user on all nodes, including the MDS. 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 wallets exist on the remote sites:
 - A wallet for storing the password to the database designated for Oracle Internet Directory. This wallet is named `oidpwd1ldap1` and is located in the directory `$ORACLE_HOME/ldap/admin`.
 - A wallet for storing the password of the replication administrator. This wallet is named `oidpwroracle_sid`, and is located in the directory `$ORACLE_HOME/ldap/admin`. (The `oracle_sid` is obtained not from the environment variable `SID` but from the connected database.)
3. Check the prerequisites in the attached Note. Then, at a command prompt in the MDS, use `remtool` (the Replication Environment Management Tool) to configure Advanced Replication by running the following script:

```
$ORACLE_HOME/bin/remtool -asrsetup
```

Note:

- The "remtool" command-line tool reference in *Oracle Identity Management User Reference* for information about using the `-asrsetup` option of the Replication Environment Management Tool (`remtool`).
 - *Oracle Database Administrator's Guide* in the Oracle Database Documentation Library for instructions on ensuring that the database and listener are running
 - *Oracle Database Net Services Administrator's Guide* in the Oracle Database Documentation Library for instructions on ensuring that the connect string is correct
-

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.

Note: As part of "[Task 3: Set Up Oracle Database Advanced Replication for a Directory Replication Group](#)", the Replication Environment Management Tool (`remtool`) sets default values for the replication configuration parameters, which enables you to simply start the replication servers. If you wish to change the replication configuration parameters, see "[Managing Replication](#)" on page 25-35.

Task 4 (Optional): Load Data into the Directory

You can choose either of two ways to load data into the directory:

- To add just a small number of entries 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.
- To add a large amount of data to load into the DRG, use the bulkload utility:

- a. Stop the LDAP server at all nodes of the DRG by typing:

```
opmnctl stopproc ias-component=OID
```

- b. On the node that is part of the DRG and where you have the `ldif` file to be loaded onto the directories enter:

```
bulkload.sh -connect connect_string -check \  
            -generate file_with_absolute_path_name
```

Note: If data is extracted from Oracle Internet Directory using `ldifwrite`, then, in addition to other options, use the `-restore` option to restore the operational attributes.

- c. On the same node, enter:

```
bulkload.sh -connect connect_string_1 -load
```

Repeat step c on the same node, 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 in the DRG. For example, enter:

```
bulkload.sh -connect connect_string_2 -load
```

then enter

```
bulkload.sh -connect connect_string_3 -load
```

and so on, until you loaded the data onto all the nodes in the DRG.

Note:

- `connect_string` is the connect string of the local Oracle Internet Directory database.
 - For successful replication, an entry must have the same `orclguid` (global identifier) at all replicated nodes. This is accomplished by performing Step b once and repeating Step c for all nodes 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 8.6 or later. Visit: <http://www.datafocus.com/>
-
-

See Also: The "bulkload" command-line tool reference in *Oracle Identity Management User Reference* for syntax and usage notes

Task 5: Ensure that Oracle Directory Server Instances are Started on All the Nodes

The out-of-box configuration has Oracle Internet Directory LDAP Server instance #1 configured with change logging set to TRUE. This default instance of Oracle Internet Directory LDAP Server can be started using `opmn` as follows:

```
opmnctl startproc ias-component=OID
```

Note: Do not use the `opmnctl startall` command at this point. If you do, the HTTP server and the Oracle Internet Directory server will start, but the command will hang while trying to start the OC4J server. The OC4J server can be started only after Task 6 is complete. Once the replication server has been started on all nodes of the DRG and replication has propagated to all nodes, you may safely use the `opmnctl startall` command.

See Also: [Chapter 3, "Post-Installation 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 need not be unique across the entire DRG.

Note: If you are deploying a single master with read-only replica consumers, you can reduce performance overhead by turning off the multimaster flag in the directory replication server. To do so, change the value of the `-m` flag in the OID Control Utility command for Oracle directory replication server from the default (`TRUE`) to `FALSE`. The multimaster option controls conflict resolution, which serves no purpose if you are deploying a single master.

Once the Oracle Internet Directory replication server has been started using `oidctl`, any `opmnctl` command to stop or start the Oracle Internet Directory component will ensure that the Oracle Internet Directory replication server process is also stopped or started.

See Also:

["Process Control of Oracle Internet Directory Components"](#) on page 4-8 for information on Oracle Internet Directory process control.

["Conflict Resolution in Multimaster Replication"](#) on page 24-19

[Chapter 5, "Oracle Directory Server Administration"](#) for information on starting the replication servers

Task 7: Test Directory Replication

Use Oracle Directory Manager to verify that the directory replication servers are running, and 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 for Multimaster Replication (Oracle Database Advanced Replication Types Only)

Note: A new node that you add to an existing multimaster replication group must have an Oracle Application Server Infrastructure installed on it. During its installation, the installation type selected had to have been **Oracle Application Server with Metadata Repository**. For more information, see "[Task 2: Install the Oracle Internet Directory as a Replica, on the Remote Master Sites \(RMS\)](#)" on page 25-6.

You can add a node to a master node, or to an LDAP-based supplier replica that is not a consumer of any other LDAP based replicas, to form a multimaster DRG. When you do so, the steps in this section will automatically perform an initial install and configuration of Advanced Replication.

To add a new replication node to a live, functioning replication group or to a master node of any significant size, perform the following steps:

- [Prepare the Oracle Net Services Environment](#)
- [Task 1: Stop the Directory Replication Server on All Nodes](#)
- [Task 2: Identify a Sponsor Node and Install Oracle Internet Directory as a Replica on the Remote Site](#)
- [Task 3: Switch the Sponsor Node to Read-Only Mode](#)
- [Task 4: Back up the Sponsor Node by Using Idifwrite](#)
- [Task 5: Perform Advanced Replication Add Node Setup](#)
- [Task 6: Switch the Sponsor Node to Updatable Mode](#)
- [Task 7: Start the Directory Replication Server on All Nodes Except the New Node](#)
- [Task 8: Load Data into the New Node by Using bulkload](#)
- [Task 9: Start the Directory Server on the New Node](#)
- [Task 10: 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 2: Identify a Sponsor Node and Install Oracle Internet Directory as a Replica on the Remote Site](#)", be sure that all three of these types of items are in the path.

Prepare the Oracle Net Services Environment

The process that prepares this environment is described in [Section , "On All Nodes, Prepare the Oracle Net Services Environment for Replication"](#).

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, type:

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

Note: The instance number might not be 1. Check the running process to discover the instance number in use here.

Task 2: Identify a Sponsor Node and Install Oracle Internet Directory as a Replica on the Remote Site

You must identify a sponsor node for this Task. It is the node that will supply the data to the new node.

For the RMS, Oracle recommends that you install the new instance of Oracle Internet Directory as an Advanced Replication replica. (You could use an existing master node as the RMS, but extra manual steps are required.)

To install a new Oracle Internet Directory as an Advanced Replication replica for RMS, use the instructions in ["If you are installing Oracle Internet Directory as a Replica"](#) on page 25-5 and choose Advanced Replication when that choice appears.

If an existing master is used as RMS, you need to follow the instructions in ["If an Existing Master is Used as a Remote Master Site"](#) on page 25-6 to migrate the master's metadata to the sponsor node. After successfully migrating the master's metadata to the MDS, you can now safely continue with ["Task 3: Switch the Sponsor Node to Read-Only Mode"](#).

Task 3: Switch the Sponsor Node to Read-Only Mode

A sponsor node is the node that supplies the data to the new node. To switch the sponsor node 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 command against the identified sponsor node:

```
ldapmodify -D "cn=orcladmin" -w adminPassword -h host_name_of_sponsor_node \  
-p port -f change_mode.ldif
```

This command changes the Oracle directory server *name_of_sponsor_node* from sponsor mode 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 4: Back up the Sponsor Node by Using `ldifwrite`

Because this may take a long time, you may start ["Task 5: Perform Advanced Replication Add Node Setup"](#) while backup is in process.

On the sponsor node, enter the following command:

```
ldifwrite -c connect_string \  
         -b "orclAgreementID=000001,cn=replication configuration" \  
         -f output_ldif_file
```

This backs up the directory of the sponsor node.

Task 5: Perform Advanced Replication Add Node Setup

Note: Oracle Net Service must be configured properly on all nodes for replication. See: "[On All Nodes, Prepare the Oracle Net Services Environment for Replication](#)" on page 25-7.

You can perform the Advanced Replication add node setup at the same time that you perform "[Task 4: Back up the Sponsor Node by Using ldifwrite](#)" on page 25-14.

On the sponsor node, enter this command:

```
remtool -addnode
```

The Replication Environment Management Tool adds the node to the DRG.

Notes:

When you run `remtool -addnode` to add the first Advanced Replication replica of a replication group, the tool does the initial replication setup for you, as if you had used `remtool -asrsetup`. You must specify the sponsor node's connect identifier when you use `remtool -addnode`.

When you use `remtool -addnode`, the operation might take a long time to complete, depending on the number of rows available in replicated tables and the network latency between the nodes. Use the `-v` option to view the progress of this operation.

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 in the list, remove the new node by running the Replication Environment Management tool with `-delnode` option. Then add the new node again using the `-addnode` option.

See Also: The "remtool" command-line reference in *Oracle Identity Management User Reference* for instructions on using the `-addnode` option of the Replication Environment Management Tool

Task 6: Switch the Sponsor Node to Updatable Mode

To switch the sponsor node to updatable mode:

1. Edit `change_mode.ldif` to the following:

```
dn:  
changetype: modify  
replace: orclservermode  
orclservermode: rw
```

2. Run the following commands on the sponsor node:

```
ldapmodify -D adminDN -w adminPassword -h host_name_of_sponsor_node \  
-p port -f change_mode.ldif
```

This command changes the Oracle directory server *host_name_of_sponsor_node* from sponsor mode back 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 7: Start the Directory Replication Server on All Nodes Except the New Node

To start the directory replication server, type the following command on all nodes except the new node:

```
oidctl connect=db_connection_string server=oidrepld instance=1 \  
flags='-h host -p port' start
```

To ensure that no directory or replication processes are running on the new node, type:

```
opmnctl stopproc ias-component=OID
```

Task 8: Load Data into the New Node by Using bulkload

To load data, type the following command on the new node:

```
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 8.6 or later. Visit: <http://www.datafocus.com/>
-
-

Task 9: Start the Directory Server on the New Node

To start the directory server, type the following command on the new node:

```
oidctl connect=db_connect_string_of_new_node server=oidldapd \  
instance=1 flags='-p port' start
```

Task 10: 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 on the new node:

```
opmnctl startproc ias-component=OID
```

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).

To delete a replication node, perform the tasks described in these topics:

- [Task 1: Stop the Directory Replication Server on All Nodes](#)
- [Task 2: Stop All Oracle Internet Directory 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=oidrep1d instance=1 stop
```

Note: The instance number may vary.

Task 2: Stop All Oracle Internet Directory Processes in the Node to be Deleted

On the node to be deleted, shut down Oracle Internet Directory using `opmn`.

```
$ORACLE_HOME/opmn/bin/opmnctl stopproc ias-component=OID
```

See Also: The "opmn" command-line tool reference in *Oracle Identity Management User Reference* for more information about shutting down Oracle Internet Directory.

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: The "remtool" command-line tool reference in *Oracle Identity Management User Reference* for instructions on using the `-delnode` option of the Replication Environment Management Tool

This process can take a long time, depending on your system resources and the size of your DRG. If you use the `-v` option, the tool 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 in the list, then delete 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 on each of the remaining nodes of the DRG:

```
oidctl connect=connect_string server=oidrepld instance=1 \
      flags='-h host -p port' start
```

See Also: The "opmn" command-line tool reference in *Oracle Identity Management User Reference*

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 Oracle Internet Directory 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 25–1 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 25–2 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 25–3 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.

Note: The Oracle Internet Directory server parameter `orclSizeLimit`, which is 1000 by default, limits the number of entries that the Human Intervention Queue Manipulation Tool can process. If you have more than 1000 entries in the human intervention queue, you must increase `orclSizeLimit`, or some entries will never be processed. Setting the parameter `orclSizeLimit` very high will impact server performance, because `orclSizeLimit` also controls the maximum number of entries to be returned by a search.

See Also: The "hiqretry.sh" command-line tool reference in *Oracle Identity Management User Reference* for instructions on how to use the Human Intervention Queue Manipulation Tool

About the Oracle Internet Directory Reconciliation Tool

When the directory replication server encounters inconsistent data, you can use the Oracle Internet Directory 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 3: Switch the Sponsor Node to Read-Only Mode"](#) for instructions on setting a node to read-only mode
- The "oidreconcile" command-line tool reference in *Oracle Identity Management User Reference* 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](#)
- [Back Up Your LDAP Data by Using Idifwrite and bulkload](#)
- [Installing and Configuring an LDAP Replica with Default Settings](#)
- [Installing and Configuring an LDAP-Based Replica with Customized Settings](#)
- [Deleting an LDAP-Based Replica](#)
- [Table , "Determining What Is to Be Replicated in LDAP-Based Partial Replication"](#)

See Also: "Supplementary Procedures for Configuring LDAP-Based Replicas" in *Oracle Application Server Administrator's Guide*

Rules for Configuring LDAP-Based Replication

The following rules apply to both full and partial LDAP-based replication:

1. An LDAP-based replica cannot have two suppliers.
2. In LDAP-based replication, only the naming contexts listed in the `namingcontexts` attribute of the root DSE can be replicated to the consumer.

See Also: The discussion of namingcontexts in

- ["Replication Fields in Oracle Directory Manager"](#) on page A-11
- ["Included and Excluded Naming Contexts"](#) on page 24-7
- ["Viewing and Modifying Replica Naming Context Objects by Using Oracle Directory Manager"](#) on page 25-32

3. The supplier of an LDAP-based replica can be a master node that is a standalone node, a member of a multimaster replication group, or another LDAP-based replica.

See Also: For instructions on installing on a standalone node, see ["If you are installing Oracle Internet Directory as a Master"](#) on page 25-4

4. An LDAP-based replica can be a consumer for another LDAP-based replica. That consumer is then called a fan-out replica.
5. You can add an Oracle Internet Directory 10g Release 2 (10.1.2) LDAP replica to an Oracle Internet Directory 10g (9.0.4) master. You can also upgrade a 10g (9.0.4) LDAP replica of a 10g (9.0.4) master to 10g Release 2 (10.1.2).

Note: Make sure the schemas are synchronized. Otherwise, the replication server might not be able to apply changes to the consumer replica.

6. The new consumer node must be empty. That is, Oracle Internet Directory must be newly installed.

Back Up Your LDAP Data by Using Idifwrite and bulkload

Use 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 a long time for a directory with one million entries.

See Also: "Oracle Identity Management Command-line Tool Reference" in *Oracle Identity Management User Reference* presents the syntax and multiple examples for both `ldifwrite` and `bulkload`.

Installing and Configuring an LDAP Replica with Default Settings

This section discusses default `namingcontext` configuration settings as one way to configure LDAP-based replication from the supplier replica to the consumer replica, with all `namingcontexts` included. That is, replication is configured so that the entire DIT of the supplier replica will be replicated to the consumer.

When you install and configure an LDAP replica with default settings, automated bootstrap performs the initial data synchronization from the supplier to the consumer. After the installation of a new LDAP replica is completed, LDAP based replication will be configured and the replication process will be started automatically.

Note: When you install an LDAP replica using default settings, the installer automatically migrates the consumer replica's metadata to the supplier replica. It is assumed all metadata are under the naming context `cn=oraclecontext`. If you need to use a different naming context, such as a realm-specific naming context, see the next section, ["Installing and Configuring an LDAP-Based Replica with Customized Settings"](#) on page 25-22.

See Also: The discussion of automatic bootstrap in [Appendix H, "LDAP Replica States"](#)

Task 1: Identify and Start the Directory Server on 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

Make sure the Oracle Internet Directory server is started on the Supplier node. To start the directory server, type the following command:

```
opmnctl startproc ias-component=OID
```

Task 2: Installing Oracle Internet Directory As An LDAP Replica

Use the instructions in ["If you are installing Oracle Internet Directory as a Replica"](#) on page 25-5 and in step 5 choose **LDAP replica**.

When you install an LDAP replica with the default settings, Oracle Universal Installer automatically invokes bootstrap to migrate data from the supplier to the consumer.

Note: Bootstrapping may take a long time to complete.

Do not update the Oracle Internet Directory schema on the supplier when bootstrapping is in progress. If you do, replication bootstrap might fail. If it fails, verify that the Oracle Internet Directory schema at the consumer is synchronized with that of the supplier before you try to bootstrap again.

If you update the supplier Oracle Internet Directory during bootstrapping, the Oracle Internet Directory replication server will log a warning message. Changes you make to the supplier will be replicated to the consumer. Some of the changes, however, might be moved to the human intervention queue.

Once installation is complete, LDAP replication will be configured. The replication server on the consumer will replicate changes from the supplier.

You can check replication activity by viewing the replication log file on the new LDAP replica.

Installing and Configuring an LDAP-Based Replica with Customized Settings

To establish customized settings, you must first install the new node as a Master node (standalone). To do so, follow the instructions in ["If you are installing Oracle Internet Directory as a Master"](#) on page 25-4.

After configuring LDAP base replication with `remtool`, you can customize the `namingcontext` defining what will be replicated for that LDAP-based node.

See Also: The discussion of naming contexts in ["Determining What Is to Be Replicated in LDAP-Based Partial Replication"](#) on page 25-32

There are two ways to install and configure an LDAP-based replica with customized setting, based on how you will migrate the data from the directory:

- Use the command-line tools. Use `ldifwrite` to backup the data from the supplier replica, then use `bulkload` to restore the data to the consumer replica
- Use automatic bootstrapping. This is a replication server feature that automatically bootstrap the data from the supplier replica to the consumer replica, based upon replication configuration.

[Table 25-1](#) compares these two methods.

Table 25-1 Data Migration Using `ldifwrite/bulkload` versus Automatic Bootstrapping

Migration Using <code>ldifwrite/bulkload</code>	Migration Using 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

If automatic bootstrapping is your chosen data migration method, customize your LDAP-based replica using "[Configuring an LDAP-Based Replica by Using Automatic Bootstrapping](#)" on page 25-23.

If `ldifwrite/bulkload` is your chosen data migration method, configure your LDAP-based replica using "[Configuring an LDAP-Based Replica by Using the `ldifwrite` Tool](#)" on page 25-27.

Configuring an LDAP-Based Replica by Using Automatic Bootstrapping

The following eight tasks enable you to configure an LDAP-based replica by using automatic bootstrapping. They are explained in the paragraphs that follow this list.

- [Task 1: Identify and Start the Directory Server on the Supplier Node](#)
- [Task 2: Create the New Consumer Node by Installing Oracle Internet Directory as a Master](#)
- [Task 3: Back Up the Metadata from the New Consumer Node](#)
- [Task 4: Add an LDAP-Based Replica by Using the Replication Environment Management Tool](#)
- [Task 5: On the Consumer, Configure the Consumer Replica for Automatic Bootstrapping](#)
- [Task 6: Optional: Change Default Replication Parameters](#)
- [Task 7: Start the Directory Replication Server on the Consumer Replica](#)
- [Task 8: If DAS or SSO Are Installed on the New Node, Restore Their Entries in the New Node's Directory](#)

Task 1: Identify and Start the Directory Server on 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

Make sure the Oracle Internet Directory server is started on the Supplier node. To start the directory server, type the following command:

```
opmnctl startproc ias-component=OID
```

Task 2: Create the New Consumer Node by Installing Oracle Internet Directory as a Master To install and configure an LDAP-based replica with customized setting, you must install the new consumer node as a master. To install a new Oracle Internet Directory as a master, follow the directions in "[If you are installing Oracle Internet Directory as a Master](#)" on page 25-4.

Task 3: Back Up the Metadata from the New Consumer Node Before configuring the new node as an LDAP-based replica with customized settings, you must first migrate its metadata to the supplier node, as follows:

- Make sure the Oracle Internet Directory server is up and running on both the supplier node and the new node created in Task 2, so that the backup process (`remtool -backupmetadata`) can succeed.
- From the newly created node, run the following command:

```
remtool -backupmetadata \  
-replica "new_node_host:new_node_port/new_node_repldn_pwd" \  
-master "master_host:master_port/master_repl_dn_pwd"
```

where *master_host:master_port/master_repldn_pwd* are the hostname, port number, and replication DN password for the desired replica's supplier.

See Also: The "remtool" command-line tool reference in *Oracle Identity Management User Reference* for more information about remtool options, including `-backupmetadata`.

- Apart from loading the metadata into master replica, this command creates a file named `ocbkup.new_replica_id.TO.master_replicaid.timestamp.dat` containing the metadata as back up. This file is created under the `$ORACLE_HOME/ldap/log` directory. This file contains the changes made to master replica in LDIF format, a copy of the SSO container entry [`orclApplicationCommonName=ORASSO_SSOSERVER, cn=SSO, cn=Products, cn=OracleContext`] and DAS URL container entry [`cn=OperationURLs, cn=DAS, cn=Products, cn=OracleContext`].
- If the metadata backup succeeds, it shows the following message in the terminal:


```
Backup of metadata will be stored in  
$ORACLE_HOME/ldap/log/ocbkup.new_replica_id.TO.master_replicaid.timestamp.dat.  
Metadata copied successfully.
```

The message will contain the path of your `ORACLE_HOME`.
- If the metadata backup is unsuccessful, the `$ORACLE_HOME/ldap/log/remtool.log` file will contain error messages. If you invoked `remtool` from a terminal, error messages appear on that terminal.

Task 4: 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 "remtool" command-line tool reference in *Oracle Identity Management User Reference* for more information about the Replication Environment Management Tool

Task 5: On the Consumer, Configure the Consumer Replica for Automatic Bootstrapping To use the automatic bootstrap capability, on the consumer, set the `orclReplicaState` attribute of the consumer replica subentry to 0 as follows:

1. Edit the sample file `mod.ldif` as follows:

```
Dn: orclreplicaid=unique_replicaID_of_consumer, cn=replication configuration
Changetype:modify
add:orclReplicaState
OrclReplicaState: 0
```

2. Use `ldapmodify` at the consumer to update the consumer replica's subentry `orclreplicastate` attribute.

```
ldapmodify -D "cn=orcladmin" -w administrator_password -h consumer_host \
-p port -f on page 24-9 mod.ldif
```

See Also: ["Managing Replication"](#) on page 25-35 for more information about the bootstrap capability of the LDAP-based replication

Task 6: Optional: Change Default Replication Parameters You can change the default parameters for replication agreements and for the replica subentry.

See Also:

- ["Replication Configuration Objects in the Directory"](#) on page 24-9
- ["Determining What Is to Be Replicated in LDAP-Based Partial Replication"](#) on page 25-32
- ["Viewing and Modifying Directory Replication Server Configuration Parameters"](#) on page 25-35
- ["Viewing and Modifying Parameters for Particular Replica Nodes"](#) on page 25-37
- ["Modifying Parameters for Replication Agreements"](#) on page 25-39

Task 7: Start the Directory Replication Server on the Consumer Replica On the consumer, start the Oracle Internet Directory replication server by typing:

```
oidctl connect=db_connect_string_of_consumer_node \
server=oidrepld instance=1 \
flags='-h consumer_host -p consumer_port' start
```

See Also: The "oidctl" command-line tool reference in *Oracle Identity Management User Reference*

When the replication server is started, it will start to bootstrap the data from the supplier to the consumer. Once the bootstrap has completed successfully, the replication server will automatically change to ONLINE mode to process changes from the supplier to the consumer.

Task 8: If DAS or SSO Are Installed on the New Node, Restore Their Entries in the New Node's Directory The entries for DAS and SSO must refer to the local instances of these services. However, the initial replication download from the supplier to the consumer creates these entries with values replicated from the supplier. If these services are in fact configured on the consumer node, then these values need to be replaced by the correct information appropriate to the consumer node.

- If the Delegated Administration Service (DAS) is configured on the consumer node, it must be restored using the following steps:

1. In the `ocbkup.new_replicaid.TO.master_replicaid.timestamp.dat` file created by Task 3, locate and copy the DAS URL. The DN of the DAS URL container entry is "cn=OperationURLs, cn=DAS, cn=Products, cn=OracleContext". It is usually the next-to-last entry in the file.

See Also: *Oracle Identity Management Guide to Delegated Administration*

2. Create an LDIF file called `change_das_url.ldif` with the following contents:

```
dn: cn=OperationURLs,cn=DAS,cn=Products,cn=OracleContext
changetype: modify
replace: orcldasurlbase
orcldasurlbase: copy_paste_the_URL_from_backup_file
```

3. Execute the following command to change the DAS URL:

```
ldapmodify -p consumer_port -h consumer_host -D super_user_DN \
-w super_user_password -f change_das_URL.ldif
```

- Similarly, if Single Sign-on (SSO) is configured on the consumer node, it must be restored using the following steps:

1. In the `ocbkup.timestamp.dat` file created by Task 3, locate and copy the SSO container entry. Copy only the attributes shown in step 2. The DN of the SSO container entry is "orclApplicationCommonName=ORASSO_SSOSERVER, cn=SSO, cn=Products, cn=OracleContext". It is usually the last entry in the file.

See Also: *Oracle Application Server Single Sign-On Administrator's Guide*

2. Create an LDIF file `add_SSO_container.ldif` with the following contents:

```
dn: orclApplicationCommonName=ORASSO_SSOSERVER,
cn=SSO,cn=Products,cn=OracleContext
orclapplicationcommonname: ORASSO_SSOSERVER
orclappfullname: ORASSO_SSOSERVER
orclversion: 10.1.2.0.0
objectclass: orclApplicationEntity
objectclass: top
userpassword: userpassword_copied_from_backup_file
```

Note: Do not copy the `authpassword;oid, createtimestamp, creatorsname, modifiersname, modifytimestamp, or orclguid` attributes.

3. Execute the following command to add the SSO container entry:

```
ldapadd -p consumer_port -h consumer_host -D super_user_DN \
-w super_user_password -f add_SSO_container.ldif
```

4. Create an LDIF file `mod.ldif` with the following contents:

```
dn: cn=OracleUserSecurityAdmins,cn=Groups,cn=OracleContext
changetype:modify
add: uniquemember
uniquemember: orclApplicationCommonName=ORASSO_SSOSEVER,
cn=SSO,cn=Products,cn=OracleContext
```

```
dn: cn=verifierServices, cn=Groups,cn=OracleContext
changetype:modify
add: uniquemember
uniquemember: orclApplicationCommonName=ORASSO_SSOSEVER,
cn=SSO,cn=Products,cn=OracleContext
```

5. Execute the following command to apply mod.ldif:

```
ldapmodify -p consumer_port -h consumer_host -D super_user_DN \
-w super_user_password -f mod.ldif
```

Restart OC4J security for the change to take effect:

```
$ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=OC4J_SECURITY
$ORACLE_HOME/opmn/bin/opmnctl startproc process-type=OC4J_SECURITY
```

6. Using a browser, test the Oracle Delegated Administration Services and OracleAS Single Sign-On pages.

To test Oracle Delegated Administration Services, try to log in as the admin user "orcladmin" on the Oracle Delegated Administration Services page, `http(s)://new_node_hostname:new_node_http_port/oiddas/`. If you cannot log in, see the troubleshooting appendix in *Oracle Identity Management Guide to Delegated Administration*.

To test OracleAS Single Sign-On, try to log in as the super admin user "orcladmin" on the OracleAS Single Sign-On page, `http(s)://new_node_hostname:new_node_http_port/pls/orasso/`. If you cannot log in, see the troubleshooting appendix in *Oracle Application Server Single Sign-On Administrator's Guide*.

Configuring an LDAP-Based Replica by Using the Idifwrite Tool

This section discusses the general tasks you perform when configuring an LDAP-based replica by using the Idifwrite tool. It contains these topics:

- [Task 1: Start the Directory Server on Both the Supplier and the Consumer Nodes](#)
- [Task 2: Back Up the Metadata from the New Consumer Node](#)
- [Task 3: Change the Directory Server at the Supplier to Read-Only Mode](#)
- [Task 4: Add an LDAP-Based Replica by Using the Replication Environment Management Tool](#)
- [Task 5: Initialize the lastappliedchangenumber Attribute](#)
- [Task 6: Back Up the Naming Contexts to Be Replicated](#)
- [Task 7: Change the Directory Server at the Supplier to Read/Write Mode](#)
- [Task 8: Load the Data on the New Consumer](#)
- [Task 9: If DAS or SSO Are Installed on the New Node, Restore Their Entries in the New Node's Directory](#)
- [Task 10: Optional: Change Default Replication Parameters](#)
- [Task 11: Start the Directory Replication Server on the Consumer Replica](#)

Task 1: Start the Directory Server on Both the Supplier and the Consumer Nodes

1. Identify the supplier for an LDAP-based replica. The supplier can be:

- standalone directory
- A node of a multi-master replication group
- Another LDAP-based replica

Make sure the Oracle Internet Directory server is started on the Supplier node. To start the directory server, type the following command:

```
opmnctl startproc ias-component=OID
```

2. Identify the consumer node, which must be an new Oracle Internet Directory install as a master. To install a new Oracle Internet Directory as a Master, follow the directions in ["If you are installing Oracle Internet Directory as a Master"](#) on page 25-4. Make sure the Oracle Internet Directory server is started on the new consumer node. To start the directory server, type the following command:

```
opmnctl startproc ias-component=OID
```

Task 2: Back Up the Metadata from the New Consumer Node Before configuring the consumer as an LDAP-based replica with customized settings, you must first migrate its metadata to the supplier node, as follows:

- Make sure the Oracle Internet Directory server is up and running on both the supplier node and the new node created in Task 2, so that the backup process (`remtool -backupmetadata`) can succeed.
- From the consumer node, run the following command:

```
remtool -backupmetadata \  
-replica "consumer_host:consumer_port/consumer_repl_dn_pwd" \  
-master "supplier_host:supplier_port/supplier_repl_dn_pwd"
```

- Apart from loading the metadata into master replica, this tool creates a file named `ocbkup.consumer_replica_id.TO.supplier_replica_id.timestamp.dat` containing the metadata as back up. This file is created in the `$ORACLE_HOME/ldap/log` directory. This file contains the changes made to the master replica in LDIF format, a copy of SSO container entry [`orclApplicationCommonName=ORASSO_SSOSERVER, cn=SSO, cn=Products, cn=OracleContext`] and DAS URL container entry [`cn=OperationURLs, cn=DAS, cn=Products, cn=OracleContext`].
- If the metadata backup succeeded, `remtool` displays the following message in the terminal:

```
Backup of metadata will be stored in  
$ORACLE_HOME/ldap/log/ocbkup.consumer_replica_id.TO.supplier_  
replicaid.timestamp.dat.  
Metadata copied successfully.
```

The message will contain the actual path of your `ORACLE_HOME` and filename.

If the metadata backup is unsuccessful, the `$ORACLE_HOME/ldap/log/remtool.log` file will contain error messages. If you invoked `remtool` from a terminal, error messages appear on that terminal.

Task 3: 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 4: 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 "remtool" command-line tool reference in *Oracle Identity Management User Reference* for more information about the Replication Environment Management Tool

Task 5: Initialize the lastappliedchangenumber Attribute To do this:

1. Search for the last applied change number on the supplier node.

```
ldapsearch -D "cn=orcladmin" -w administrator_password -h supplier_host \
-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. On the supplier, modify the agreement by using ldapmodify:

```
ldapmodify -D "cn=orcladmin" -w password -h host_name -p port_number \
-f LDIF_file
```

Task 6: 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 4: Add an LDAP-Based Replica by Using the Replication Environment Management Tool](#)" on page 25-29.

```
ldapsearch -h supplier_host -p port \
-b "orclreplicaid=supplier_replicaID,cn=replication configuration" \
-s sub "(orclreplicadn= orclreplicaid=consumer_replica_ID, \
cn=replication configuration)" dn
```

2. On the supplier, 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_retrieved_in_step_1" \  
          -f name_of_output_LDIF_file
```

See Also:

["Determining What Is to Be Replicated in LDAP-Based Partial Replication"](#) on page 25-32

The "ldifwrite" command-line tool reference in *Oracle Identity Management User Reference* for more instructions on using ldifwrite to back up part of the naming context

Task 7: Change the Directory Server at the Supplier to Read/Write Mode If you performed ["Task 3: Change the Directory Server at the Supplier to Read-Only Mode"](#) on page 25-28, 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. Run the following command:

```
ldapmodify -D "cn=orcladmin" -w administrator_password \  
          -h host_name_of_supplier_node -p port -f name_of_LDIF_file
```

Task 8: Load the Data on the New Consumer 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 -base "naming_context"
```

Perform this step for each naming context that was backed up in ["Task 6: Back Up the Naming Contexts to Be Replicated"](#) on page 25-29.

On the consumer, 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 -restore backup_data.ldif  
bulkload.sh -connect connect_string_of_replica -load
```

See Also: ■ The "bulkload" command-line tool reference in *Oracle Identity Management User Reference* for instructions on using bulkload in either the default mode or the append mode

- The "bulkdelete" command-line tool reference in *Oracle Identity Management User Reference*

Task 9: If DAS or SSO Are Installed on the New Node, Restore Their Entries in the New Node's Directory Follow the procedure described in ["Task 8: If DAS or SSO Are Installed on the New Node, Restore Their Entries in the New Node's Directory"](#) on page 25-25.

Task 10: Optional: Change Default Replication Parameters You can change the default parameters for replication agreements, for the replica subentry, and for the replication naming context configuration objects.

See Also:

- ["Viewing and Modifying Directory Replication Server Configuration Parameters"](#) on page 25-35
- ["Viewing and Modifying Parameters for Particular Replica Nodes"](#) on page 25-37
- ["Modifying Parameters for Replication Agreements"](#) on page 25-39
- ["Replication Configuration Objects in the Directory"](#) on page 24-9
- ["Determining What Is to Be Replicated in LDAP-Based Partial Replication"](#) on page 25-32

Task 11: Start the Directory Replication Server on the Consumer Replica Start the Oracle directory replication server, following the procedure described in the "oidctl" command-line tool reference in *Oracle Identity Management User Reference*.

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: Delete the Replica from the Replication Group](#)
- [Task 3: Stop the Directory Server on the Node to be Deleted](#)

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

Stop the Oracle directory replication server, following the procedure described in the "oidctl" command-line tool reference in *Oracle Identity Management User Reference*.

Task 2: 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]
```

See Also: The "remtool" command-line tool reference in *Oracle Identity Management User Reference*

Task 3: Stop the Directory Server on the Node to be Deleted

See Also: The "oidctl" command-line tool reference in *Oracle Identity Management User Reference*

Determining What Is to Be Replicated in LDAP-Based Partial Replication

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 A-20, "Fields in the Replica Naming Context Tab Page"](#).
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 A-20, "Fields in the Replica Naming Context Tab Page"](#).
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 Idapmodify

Replica naming context object parameters are listed and described under Replication Schema Elements in *Oracle Identity Management User Reference*.

Note: The replication server reads naming context objects from the supplier replica.

Example 25–4 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=supplier_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 the supplier.

```
ldapadd -D "cn=orcladmin" -w administrator_password -h supplier_host \
-p port_number -f mod.ldif
```

Example 25–5 Deleting a Naming Context Object

To delete the naming context object created in [Example 25–4](#), type:

```
ldapdelete -D "cn=orcladmin" -w administrator_password \
-h supplier_host -p supplier_host_port_number \
"cn=naming_context_identifier,cn=replication namecontext, \
orclagreementid=replication_agreement_identifier, \
orclreplicaid=supplier_replica_identifier, \
cn=replication configuration"
```

Example 25–6 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 `c=us` is to be included in partial replication.

1. Edit the example file `mod.ldif` as follows:

```
DN:cn=naming_context_identifier,cn=replication namecontext,
    orclagreementid=replication_agreement_identifier,
    orclreplicaid=supplier_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 supplier_host \
-p port -f mod.ldif
```

3. Restart the directory replication server.

Example 25–7 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:

```
DN:cn=naming_context_identifier,
cn=replication namecontext,
orclagreementid=replication_agreement_identifier,
orclreplicaid=supplier_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 supplier_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 25–8 Modifying the `orclExcludedAttributes` Attribute for a Replica Naming Context Object

You can specify that certain changes made to the included naming context be excluded, at attribute level, 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:

```
DN:cn=naming_context_identifier,
cn=replication namecontext,
orclagreementid=replication_agreement_identifier,
orclreplicaid=supplier_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 24-8
- ["The Replica Subentry"](#) on page 24-10

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

The section "Directory Server Configuration Parameters" under "Replication Schema Elements" in *Oracle Identity Management User Reference* 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=osdrep1d,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.

For a description of these fields, see [Table A-16, "Fields in the Replication Server Configuration Set: General Tab Page"](#).

4. For Advanced Replication-based agreements, in the **ASR Agreement** tab page, modify the fields as appropriate.

For a description of these fields, see [Table A-17, "Fields in the ASR Agreement 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 the "ldapmodify" command-line tool reference in *Oracle Identity Management User Reference*.

"Replication Schema Elements" in *Oracle Identity Management User Reference* describes the replication server configuration parameters. As noted in that table, the modifiable replication configuration parameters are:

- `orclChangeRetryCount`
- `orclThreadsPerSupplier`

Example 25-9 *Modifying the Number of Retries Before a Change Is Moved into the Purge Queue*

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=osdrep1d,cn=subconfigsentry
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 port -f mod.ldif
```

3. Restart the directory replication server.

Example 25–10 Modifying the Number of Worker Threads Used in Change Log Processing

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=osdrep1d,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 port -f mod.ldif
```

3. Restart the directory replication server.

See Also: The "oidctl" command-line tool reference in *Oracle Identity Management User Reference* 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. "Replication Schema Elements" in *Oracle Identity Management User Reference* 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 24-10 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 A–18, "Fields in the Replica Node: General Tab Page"](#) 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 A-19, "Columns in the Replica Agreements Tab Page"](#).

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 the "ldapmodify" command-line tool reference in *Oracle Identity Management User Reference*.

Example 25-11 Modifying the orclReplicaURI Attribute for a Particular Replica Node

The directory replication server uses the `orclReplicaURI` 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 port -f mod.ldif
```

3. Restart the directory replication server.

Example 25-12 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 port -f mod.ldif
```

3. Restart the directory replication server.

Example 25-13 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_replicaID, cn=replication configuration
```

```

Changetype:modify
replace:orclReplicaState
OrclReplicaState: 0

```

2. On the host where the consumer replica is running, use `ldapmodify` to update the replica subentry `orclreplicastate` 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

This section contains instruction for modifying replication agreements that are based on both Advanced Replication and LDAP.

Modifying Parameters for Replication Agreements Based on Oracle Database Advanced Replication

Replication agreement parameters based on Advanced Replication are stored in replication agreement entries, which have the following DN:

```
orclAgreementID=000001,cn=replication configuration
```

Note:

- For replication agreements based on 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 10g Release 2 (10.1.2), only one replication agreement based on Oracle Database 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 Oracle Database Advanced Replication by Using Oracle Directory Manager" on page 25-39](#)
- ["Managing Replication Agreements Based on Advanced Replication by Using `ldapmodify`" on page 25-40](#)

Viewing and Modifying Replication Agreements Based on Oracle Database 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 **Oracle Internet Directory Servers**, then **directory server instance**, then **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 an Advanced Replication-based replication agreement. The fields in this tab page are described in [Table A-17, "Fields in the ASR Agreement Tab Page"](#).

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 are modifying the values, and 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 Advanced Replication by Using ldapmodify
 "Replication Schema Elements" in *Oracle Identity Management User Reference* 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 25-14 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 `orclagreementid=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 for Multimaster Replication \(Oracle Database Advanced Replication Types Only\)"](#) on page 25-13 explains the process of adding a new node to a replication environment.

Because Oracle Internet Directory 10g Release 2 (10.1.2) supports only one configuration set for the directory replication server, you do not need to specify a configuration set.

Example 25–15 Modifying the `orclExcludedNamingContexts` Attribute for an Oracle Database Advanced Replication Replica Agreement

In a replication agreement based on 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 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=unique_identifier_of_the_replication_agreement,
orclReplicaId=unique_identifier_of_the_supplier, cn=replication configuration
```

Note: Ensure that the agreement is identical at both the supplier and the consumer. The replication server reads the last applied change number and the naming context from the agreement at the supplier node. It reads the other agreement attributes 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 A-19, "Columns in the Replica Agreements Tab Page"](#).
 - **Replica Naming Context**, in which you can view, add, delete, and modify LDAP naming context objects. The fields in this tab page are described in [Table A-20, "Fields in the Replica Naming Context Tab Page"](#).

Modifying LDAP-Based Replication Agreement Parameters by Using `ldapmodify` "Replication Schema Elements" in *Oracle Identity Management User Reference* describes the replication agreement parameters and indicates those that you can modify.

Example 25-16 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 25-17 Modifying the `orclLastAppliedChangeNumber` Attribute for a 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 the partial replication add node procedure.

1. Edit the example file `mod.ldif` as follows:

```
dn: orclAgreementID=unique_identifier_of_the_replication_agreement,
   orclReplicaID=unique_identifier_of_the_supplier,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` argument to the Replication Environment Management Tool, `remtool`. To use this argument, enter:

```
remtool -chgpwd
```

The `remtool` utility then 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.

You can also use the `-pchgpwd` argument to `remtool` to change the password of the replication DN of a replica.

To change the password only in the replication wallet, `$ORACLE_HOME/dap/admin`, use the `-pchgwlpwd` argument to `remtool`. To use this argument, enter:

```
remtool -pchgwlpwd
```

See Also: The "remtool" command-line tool reference in *Oracle Identity Management User Reference* 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 to 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 A-21, "Fields in the Change Log Window"](#).

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 Oracle Database Advanced Replication

In directory replication based on 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 Advanced Replication to push the change to the consumer
- 1 minute for the consumer to apply the change

In the case of 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 -D "cn=orcladmin" -w administrator_password -h host_name -p port \
-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=unique_identifier_of_the_replication_agreement,
orclReplicaId=unique_identifier_of_the_supplier,
cn=replication configuration
changetype:modify
replace: orclupdateschedule
orclupdateschedule: 0
```

2. On the consumer host, upload `mod.ldif` as follows:

```
ldapmodify -h consumer_host_name -p consumer_port -D cn=orcladmin \
-w administrator_password -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 four systems as described in [Table 25-2](#).

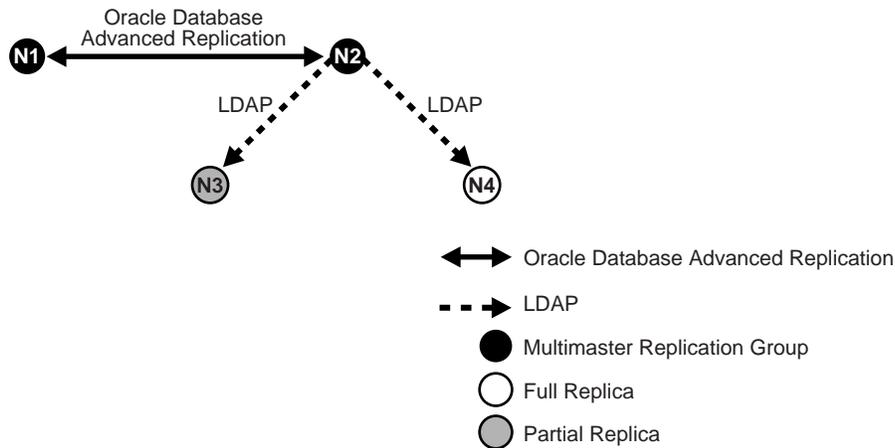
Table 25-2 *Nodes in Example of Partial Replication Deployment*

Node	Host Name	Port
Node1	mycompany1.com	3000
Node2	mycompany2.com	4000
Node3	mycompany3.com	5000
Node4	mycompany4.com	6000

In this example, the user has set up the following requirements:

- **Requirement 1:** 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.
- **Requirement 2:** 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`.
- **Requirement 3:** Node4 is to be configured as a full replica of node 2, that is, changes to all naming contexts in Node2 will be replicated (one-way) to Node4.

Figure 25–1 Example of Fan-Out Replication



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, and for the third, full LDAP replication from Node2 to Node4.

This section contains these topics:

- [Task 1: Set up the Multimaster Replication Group for Node1 and Node2](#)
- [Task 2: Configure the Replication Agreement](#)
- [Task 3: Start the Replication Servers on Node1 and Node2](#)
- [Task 4: Test the Directory Replication Between Node1 and Node2.](#)
- [Task 5: Install and Configure Node3 as a Partial Replica of Node2](#)
- [Task 6: Customize the Partial Replication Agreement](#)
- [Task 7: Start the Replication Servers on All Nodes in the DRG](#)
- [Task 8: Install and Configure Node4 as a Full Replica of Node2](#)
- [Task 9: Test the Replication from Node2 to Node4](#)

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 "[Installing and Configuring a Multimaster Replication Group](#)" on page 25-3.

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 25-11.

Task 4: Test the Directory Replication Between Node1 and Node2.

To do this, follow the instructions in "[Task 7: Test Directory Replication](#)" on page 25-12.

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 5 in "[Configuring an LDAP-Based Replica by Using Automatic Bootstrapping](#)" on page 25-23.

If you want to configure the replica by using the `ldifwrite` tool, then follow Tasks 1 through 9 in "[Configuring an LDAP-Based Replica by Using the `ldifwrite` Tool](#)" on page 25-27.

Identify Node2 as the supplier and Node3 as the consumer.

Task 6: Customize the Partial Replication Agreement

To do this:

- To achieve Requirement 2 in this example, the default replication between Node2 and Node3 must be configured first:

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 (Node2, `mycompany2.com`) and the consumer (Node3, `mycompany3.com`).

```

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:

- a. 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
```

- b. 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 edit the example file `mod.ldif` and use `ldapmodify` to modify the partial replica `orclreplicastate` attribute at both Node2 and Node3, as described in ["Task 5: On the Consumer, Configure the Consumer Replica for Automatic Bootstrapping"](#) on page 25-25.

Task 7: Start the Replication Servers on All Nodes in the DRG

To do this, follow the instructions in ["Task 11: Start the Directory Replication Server on the Consumer Replica"](#) on page 25-31.

Task 8: Install and Configure Node4 as a Full Replica of Node2

Since full replica replication is the default configuration when the new node is installed as an LDAP replica, use the instructions in ["Installing and Configuring an LDAP Replica with Default Settings"](#) on page 25-21. When the installer prompts for the supplier information, provide the supplier hostname, `mycompany2.com`, the supplier port, 4000, and the super user password.

Note: While installing node 4 as described in ["If you are installing Oracle Internet Directory as a Replica"](#) on page 25-5, you will be asked to provide the supplier's hostname, port, and `administrator_user_password`. For example:

```
Hostname: mycompany2.com  
Port:4000  
Administrator_user_password: admin_user_password
```

Task 9: Test the Replication from Node2 to Node4

Test this replication using the instructions in the section entitled ["Task 7: Test Directory Replication"](#) on page 25-12.

Part VI

Directory Plug-ins

This part contains these chapters:

- [Chapter 26, "Oracle Internet Directory Plug-in Framework"](#)
- [Chapter 27, "Oracle Internet Directory Plug-In for Password Policies"](#)
- [Chapter 28, "Setting Up the Customized External Authentication Plug-in"](#)

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](#)
- [Creating Plug-ins](#)
- [Registering and Managing Plug-ins](#)

See Also: The chapter on the Oracle Internet Directory server plug-in framework in *Oracle Identity Management Application Developer's Guide*.

About Directory Server Plug-ins

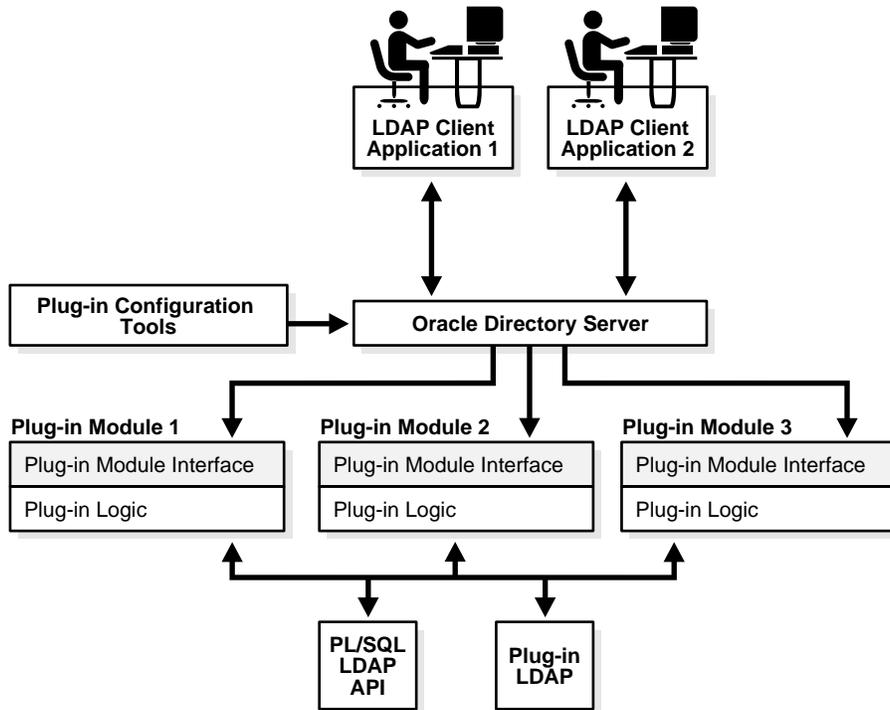
Directory server plug-ins can provide the directory server with the following kinds of added functionality, to mention just a few:

- 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 26-1](#), 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 26–1 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 26–1](#) explains the various kinds of operation-based plug-ins.

Table 26–1 Types of Operation-Based Plug-ins

Type of Plug-in	Description
Pre-operation	Plug-ins that the directory server calls <i>before</i> performing an LDAP operation. Typically, these plug-ins validate data before using it in an LDAP operation. If validation fails, then depending on the error or warning returned from the plug-in, the LDAP operation can decide to proceed or not. However, if the associated LDAP request fails later on, then Oracle Internet Directory does not roll back whatever the plug-in has already committed.
Post-operation	Plug-ins that the directory server calls <i>after</i> 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.

Table 26–1 (Cont.) Types of Operation-Based Plug-ins

Type of Plug-in	Description
When-operation	<p>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.</p> <p>There are different types of When-operation plug-ins—namely, Add-on and Replace.</p> <p>The Add-on plug-in can perform <code>ldapadd</code>, <code>ldapdelete</code>, and <code>ldapmodify</code> operations.</p> <p>The Replace plug-in can perform <code>ldapcompare</code>, <code>ldapbind</code>, and <code>ldapmodify</code> operations.</p> <p>For example, for the <code>ldapcompare</code> 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.</p>

Creating Plug-ins

Creating a plug-in module is like creating a PL/SQL package. Both have a specification part and a body part. The directory, not the plug-in, defines the plug-in specification because the specification serves as the interface between Oracle Internet Directory and the custom plug-in.

For security reasons and for the integrity of the LDAP server, you can compile plug-ins only in the ODS database schema. You must compile them in the database that serves as the backend database of Oracle Internet Directory.

See Also: *Oracle Identity Management Application Developer's Guide* for more information.

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" in *Oracle Identity Management User Reference* 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 **Oracle Internet Directory Servers**, then ***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 A-15](#) on page A-9.
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 **Oracle Internet Directory Servers**, then ***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 A-15](#) on page A-9.
5. Choose **OK**.

Deleting a Plug-in by Using Oracle Directory Manager

To delete a plug-in:

1. In the navigator pane, expand **Oracle Internet Directory Servers**, then ***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" in *Oracle Identity Management User Reference* 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.

Note: 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=subconfigsentry"
```

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](#)
- [Example: Installing, Configuring, and Enabling a Customized Password Policy Plug-in](#)

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 a password policy plug-in could perform include:

- 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
- Whether it is a dictionary word

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 27-4. 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 and that it does not contain a dictionary word that is longer than four 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. Also, if you try to add or modify a user with the `userpassword` `supersunday`, the password is rejected because `super` and `sunday` are dictionary words.

The dictionary is a list of words longer than four characters, initially stored in a file called `words.txt`. Before we implement the plug-in, we set up a database table and store the words into the table. To set up the table we use `create.sql`, which has the following contents:

```
drop table mydic;
create table mydic (word varchar2(1024));
commit;
exit;
```

Then we load the words into the table using the `sqlldr` command:

```
sqlldr control=words.txt userid=ods/ods_password
```

This section contains these topics:

- [Loading and Registering the PL/SQL Program](#)
- [Coding the Password Policy Plug-in](#)

- [Debugging the Password Policy Plug-in](#)
- [Contents of Sample PL/SQL Package pluginpkg.sql](#)

Loading and Registering the PL/SQL Program

Once you have 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=subconfigsentry
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

### modify plugin ###
dn: cn=pre_mod_plugin,cn=plugin,cn=subconfigsentry
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: 1. check if the password is at least
--          8 characters long
--          2. check if the password contains a
--          dictionary word (longer than 4 characters)

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;
    lcount   NUMBER DEFAULT 0;
    matched  VARCHAR2(1024) DEFAULT NULL;

    CURSOR c1 IS
    SELECT word FROM mydic WHERE length(word) > 4
    AND instr(lower(inpwd), lower(word), 1, 1) > 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
        OPEN c1;
        LOOP
            FETCH c1 INTO matched;
            EXIT WHEN c1%NOTFOUND;
            lcount := lcount + 1;
        END LOOP;
        plg_debug( 'count = ' || lcount);
        IF lcount > 0 THEN
            RETURN 2;
        ELSE
            RETURN ret;
        END IF;
    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,
    errmsg  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;
        errmsg := 'PASSWORD POLICY VIOLATION:0000X, less than 8 chars';
        plg_debug( ' we got an invalid password, too short ');
        ELSIF (ret = 2) THEN
            rc := 1;
            errmsg := 'PASSWORD POLICY VIOLATION:0000X, contains dictionary word';
            plg_debug( ' we got an invalid password, dictionary word ');
        ELSE
            plg_debug( ' we got a good password ');
            rc := 0;
            errmsg := '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;
        errmsg := 'exception: pre_add plguin';
END;

PROCEDURE pre_modify (ldapplugincontext IN ODS.plugincontext,
    dn          IN VARCHAR2,
    mods       IN ODS.modlist,
    rc         OUT INTEGER,
    errmsg     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, too short ');
  ELSIF (ret = 2) THEN
    rc := 1;
    errormsg := 'PASSWORD POLICY VIOLATION:0000X, contains dictionary word';
    plg_debug( ' we got an invalid password, dictionary word ');
  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

EXIT;
```

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](#)
- [Example: Installing, Configuring, and Enabling the External Authentication Plug-in](#)

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](#)
- [Debugging the External Authentication Plug-in](#)
- [Contents of PL/SQL Package oidexaup.sql](#)

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 28-4. 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`
- One utility function —namely, `get_nickname`

The integrated package is the plug-in package, `OIDEXTAUTH`. It can also serve as a template to modify according to the requirements of your deployment.

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=subconfigsentry
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=subconfigsentry
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 the 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_HOME/ldap/admin/oidspdsu.sql
```

To enable directory server plug-in debugging, execute the following command:

```
sqlplus ods/password @$ORACLE_HOME/ldap/admin/oidspdon.sql
```

To disable directory server plug-in debugging, execute the following command:

```
sqlplus ods/password @$ORACLE_HOME/ldap/admin/oidspdof.sql
```

To show directory server plug-in debugging messages, execute the following command:

```
sqlplus ods/password @$ORACLE_HOME/ldap/admin/oidspdsh.sql
```

To delete directory server plug-in debugging messages, please execute the following command:

```
sqlplus ods/password @$ORACLE_HOME/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 OIEXTAUTH 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 OIEXTAUTH;
/

SHOW ERROR

CREATE OR REPLACE PACKAGE BODY OIEXTAUTH 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;
    END;

```

```

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
            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;
        errmsg := '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,
                                errmsg           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;

```

```
        END;
    ELSE
        rc := 1;
        errmsg := 'PLG_Exception. Not enough info to change passwd.';
    END IF;

    plg_debug( 'external change password succeed');
    rc := 0;
    errmsg := '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;
        errmsg := 'PLG_Exception: when_modify_replace plguin';
        plg_debug('Exception in when_modify. Error code is ' || to_char(sqlcode));
        plg_debug(' ' || Sqlerrm);
    END;

END OIEXTAUTH;
/
SHOW ERRORS
--list

EXIT;
```


Part VII

Appendixes

This part contains these appendixes:

- [Appendix A, "Windows and Fields in Oracle Directory Manager"](#)
- [Appendix B, "The LDAP Filter Definition"](#)
- [Appendix C, "The Access Control Directive Format"](#)
- [Appendix D, "Globalization Support in the Directory"](#)
- [Appendix E, "Setting up Access Controls for Creation and Search Bases for Users and Groups"](#)
- [Appendix F, "The Multimaster Replication Process"](#)
- [Appendix G, "Searching the Directory for User Certificates"](#)
- [Appendix H, "LDAP Replica States"](#)
- [Appendix I, "Addition of a Directory Node by Using the Database Copy Procedure"](#)
- [Appendix J, "Troubleshooting Oracle Internet Directory"](#)

Windows and Fields in Oracle Directory Manager

This appendix lists and describes the various windows and fields Oracle Directory Manager. It contains these topics:

- [Connection Management Fields in Oracle Directory Manager](#)
- [Access Control Management Fields in Oracle Directory Manager](#)
- [Attribute Uniqueness Fields in Oracle Directory Manager](#)
- [Garbage Collection Management Fields in Oracle Directory Manager](#)
- [Oracle Internet Directory Statistics Collection Management Fields in Oracle Directory Manager](#)
- [Password Policy Fields in Oracle Directory Manager](#)
- [Password Verifier Fields in Oracle Directory Manager](#)
- [Plug-in Management Fields in Oracle Directory Manager](#)
- [Replication Fields in Oracle Directory Manager](#)
- [Schema Management Fields in Oracle Directory Manager](#)
- [Server Management Fields in Oracle Directory Manager](#)
- [SSL Management Fields in Oracle Directory Manager](#)
- [Synchronization Fields in Oracle Directory Manager](#)

Connection Management Fields in Oracle Directory Manager

Table A-1 Fields in the Credentials Tab Page

Field	Description
User	<p>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.</p> <p>If you are logging in as the super user, in the User box, type <code>cn=orcladmin</code>.</p> <p>If you are logging in anonymously, leave the User box empty.</p> <p>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:</p> <ul style="list-style-type: none"> ▪ 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, <ul style="list-style-type: none"> <code>cn=Susie Brown,ou=HR,o=acme,c=us</code>
Password	<p>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, <code>welcome</code>. After you are logged into Oracle Directory Manager and have connected to a directory server, you should change this password to protect the directory.</p> <p>If you are logging in anonymously, leave the Password field empty.</p> <p>If you want to login as a specific directory user, enter the corresponding password.</p> <p>See Also: "Managing Super Users, Guest Users, and Proxy Users" on page 5-8 for instructions on how to change the password</p>
Server	<p>From the Server list, select the host containing the directory server to which you want to connect.</p> <p>If you are already connected to a directory server, and you want to connect to one on a different host:</p> <ol style="list-style-type: none"> 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. <p>To add a directory server to the list:</p> <ol style="list-style-type: none"> 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. <p>To modify a directory server on the list:</p> <ol style="list-style-type: none"> 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.

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

Field	Description
Port	<p>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.</p> <p>To change this port number:</p> <ol style="list-style-type: none"> 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.
SSL Enabled	<p>Selecting this check box causes all commands you issue by using Oracle Directory Manager to be sent over Secure Sockets Layer (SSL).</p> <p>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.</p> <p>You can connect in this way if both of the following two conditions are met:</p> <ul style="list-style-type: none"> ■ 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.

Table A–2 Fields in the SSL Tab Page

Field	Description
SSL Location	<p>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:</p> <pre>file: absolute_path_name</pre> <p>If the wallet is on another machine, then link to that location and enter the linked path and file name of the wallet.</p>
SSL Password	The password to open the user's wallet
SSL Authentication	<p>Select the authentication level:</p> <ul style="list-style-type: none"> ■ 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.

Access Control Management Fields in Oracle Directory Manager

Table A-3 *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 A-4](#) lists and describes the authentication choices—that is, the methods by which users can be authenticated to the directory.

Table A-4 *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
Proxy	<ul style="list-style-type: none"> Binding as a proxy user. Specifying this authentication option blocks anonymous access.
Simple	<ul style="list-style-type: none"> Password-based authentication. Specifying this option blocks both Proxy and Anonymous access.

[Table A-5](#) lists and describes the encryption choices—that is, the method by which data is encrypted.

Table A-5 *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 14-8

Table A-6 *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.

Table A–6 (Cont.) Entities to Whom You Are Granting Access in the By Whom Tab Page

Entity	Description
When Session User's Group Is Identified By Attribute	Any group whose DN is an attribute in the entry.
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 A–7 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 DN's 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: <pre>access to attr=(member) by dnattr=(member) (selfwrite)</pre> 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 A–8 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.
Unique Attribute Scope	The filter you want the directory server to use when searching for an attribute constraint. For example: <ul style="list-style-type: none"> ▪ 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 A–9 *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 <i>YYYYMMDDHH24MISS</i> . This attribute is optional. The default value is 0, which means that the garbage collector is enabled immediately.
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.

Oracle Internet Directory Statistics Collection Management Fields in Oracle Directory Manager

The Oracle Internet Directory statistics collector supports three of the modifiable fields described in [Table A–9](#). They are **Purge Interval**, **Purge Now**, and **Purge Start**.

Password Policy Fields in Oracle Directory Manager

Table A–10 *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.

Table A–10 (Cont.) Fields in the Password Policies General Tab Page

Field	Description
Reset password upon next login	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.
Password Expiration Warning	<p>Enter the number of seconds in which users must modify their passwords before those passwords expire.</p> <p>The directory server sends a password expiration warning if these two conditions are met:</p> <ul style="list-style-type: none"> ■ The attribute for the expiry time for a user's password is set ■ This attribute is also enabled <p>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.</p> <p>For example, suppose that:</p> <ul style="list-style-type: none"> ■ 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 <p>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.</p> <p>For this feature to work, the client application must support it.</p> <p>The default is 0, which means no warnings are sent.</p>
Password Expiry Time	<p>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.</p> <p>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.</p>
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 A–11 *Fields in the Password Policies Account Lockout Tab Page*

Field	Description
Global Lockout Duration	<p>Enter the number of seconds a user is locked out of the global directory if both of the following are true:</p> <ul style="list-style-type: none"> ■ Global lockout is enabled ■ The user has been unable to bind successfully to the directory for at least the number of times specified by <code>pwdMaxFailure</code> <p>You can set user lockout for a specific duration, or until the administrator resets the user's password. The default value is 24 hours. A user account stays locked even after the lockout duration has passed unless the user binds with the correct password.</p>
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 A–12 *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. A user account stays locked even after the lockout duration has passed unless the user binds with the correct password.
IP Lockout Maximum Failure	Specify the maximum number of failed logins from a specific IP address after which the account is locked.

Table A–13 *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.
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 A–14 *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.

Table A–14 (Cont.) Fields in the Password Verifier Profile Dialog Box

Field	Description
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	<p>Parameters containing information for generating this password verifier. Use this field to specify the hashing algorithm for this password verifier. The syntax is:</p> <pre>crypto:hashing_algorithm</pre> <p>For example, if you are using the ORCLLM hashing algorithm, then you would enter:</p> <pre>crypto:ORCLLM</pre> <p>If you are using SASL/MD5, for example, you can enter the following:</p> <pre>crypto:SASL/MD5 \$ realm:dc=com</pre>

Plug-in Management Fields in Oracle Directory Manager

See Also: ["Registering and Managing Plug-ins"](#) on page 26-3

Table A–15 Fields in the New Plug-in Dialog Box

Field	Description
Mandatory Properties Tab Page	
Plug-in Enable	<p>Acceptable values are:</p> <ul style="list-style-type: none"> ■ Disable (default) ■ Enable <p>This attribute is optional.</p>
Plug-in Entry Name	For example, cn=my_plugin. This field is mandatory.
Plug-in Replacement	<p>For WHEN timing plug-in only. Possible values are:</p> <ul style="list-style-type: none"> ■ Disable (default) ■ Enable <p>This property can be enabled only if the Plug-in LDAP Operation property is ldapbind, ldapcompare, or ldapmodify.</p> <p>This attribute is optional.</p>
Plug-in Kind	PL/SQL. This field is mandatory.
Plug-in LDAP Operation	<p>One of the following values:</p> <ul style="list-style-type: none"> ■ ldapcompare ■ ldapmodify ■ ldapbind ■ ldapadd ■ ldapdelete ■ ldapsearch <p>This field is mandatory.</p>

Table A–15 (Cont.) Fields in the New Plug-in Dialog Box

Field	Description
Plug-in Package Name	This field is mandatory.
Plug-in Timing	<p>One of the following values:</p> <ul style="list-style-type: none"> ▪ pre--for plug-ins that the directory server calls <i>before</i> 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 <p>This attribute is optional.</p>
Plug-in Type	<p>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.</p> <p>This field is mandatory.</p> <p>See Also: Chapter 26, "Oracle Internet Directory Plug-in Framework"</p>
Optional Properties Tab Page	
Plug-in Attribute List	A list of semicolon-separated attribute names that controls whether the plug-in takes effect. If the target attribute is included in the list, then the plug-in is invoked.
Plug-in Entry Properties	An LDAP search filter type. For example, if you specify <code>orclPluginEntryProperties:(objectclass=inetorgperson)(sn=Cezanne)</code> , then the plug-in will not be invoked if the target entry has <code>objectclass</code> equal to <code>inetorgperson</code> and <code>sn</code> equal to <code>Cezanne</code> .
Plug-in Request Group	<p>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.</p> <p>For example, if you specify <code>cn=security,cn=groups,dc=oracle,dc=com</code>, 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 <code>cn=security,cn=groups,dc=oracle,dc=com</code>.</p>
Plug-in Result Code	<p>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.</p> <p>This is only for the POST plug-in type.</p>
Plug-in Subscriber DN List	<p>A semicolon separated DN list that controls if the plug-in takes effect. For example:</p> <pre>orclPluginSubscriberDNList=dc=COM,c=us; dc=us,dc=oracle,dc=com;dc=org,dc=us;o=IMC,c=US</pre> <p>The target DN of an LDAP operation is included in the list, then the plug-in is invoked.</p>
Plug-in Version	Supported plug-in version number. This attribute is optional.

Replication Fields in Oracle Directory Manager

Table A–16 *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 A–17 *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: <ul style="list-style-type: none"> ■ ODS_ASR_1.0 (Advanced Replication-based replication) ■ ODS_LDAP_1.0 (LDAP-based replication)
Replication Group Nodes	For Advanced Replication-based groups, enter the <code>orclreplicaid</code> 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 A–18 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 <code>orclreplicaID</code> attribute at the root DSE. You cannot modify this value.
Replica Secondary URI	Contains the set of <code>ldapURI</code> format addresses that can be used if the <code>orclReplicaURI</code> values cannot be used.
Replica State	Defines the state of the replica such as bootstrap, online, and so on. Possible values: <ul style="list-style-type: none"> ▪ 0 (Boot Strapping) ▪ 1 (On-line) ▪ 2 (Off-line) ▪ 3 (Bootstrap in progress) ▪ 4 (Bootstrap in progress, <code>cn=oraclecontext</code> bootstrap has completed) ▪ 5 (Bootstrap completed, failure detected for one or more naming contexts)
Replica Type	Defines the type of replica such as read-only or read/write. Possible values: <ul style="list-style-type: none"> ▪ 0 (Read/Write) ▪ 1 (Read-Only)
Replica URI	Contains information in <code>ldapURI</code> 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 A–19 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 Advanced Replication-based agreements.
Replica Agreement ID	Naming attribute for the replication agreement entry.

Table A–19 (Cont.) Columns in the Replica Agreements Tab Page

Column	Description
Replication Protocol	This attribute defines the replication protocol for change propagation to the replica. Values: <ul style="list-style-type: none"> ■ ODS_ASR_1.0 (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 A–20 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.
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 <code>orclincludednamingcontexts</code> 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 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 <code>orclxcludednamingcontexts</code> 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 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 25-32

Table A–21 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

Table A–21 (Cont.) Fields in the Change Log Window

Field	Description
Change Retry Count	The number of attempts to apply this change to another node in a replicated environment
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 8, "Directory Schema Administration"](#)

This section contains these topics:

- [Object Classes Fields in Oracle Directory Manager](#)
- [Attributes Fields in Oracle Directory Manager](#)
- [Matching Rules Fields in Oracle Directory Manager](#)
- [Content Rules Management Fields in Oracle Directory Manager](#)

Object Classes Fields in Oracle Directory Manager

Table A–22 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 <code>Name Exact Match subAcl</code> gives you the <code>subAcl</code> object class.
Object ID	Object identifier for the object class for which you are searching. For example, the phrase <code>Object ID Begins With 2.5.2</code> gives you a list of object classes whose object identifiers begin with <code>2.5.2</code> . 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.
Description	Word in the description field. For example, the phrase <code>Description Contains Shoe</code> gives you a list of object classes with the word <i>shoe</i> 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 <code>Add</code> displays the <code>Super Class Selector</code> dialog box from which you can select the superclass(es) you want to add.

Table A–22 (Cont.) Object Class Properties Listed in Searches in Oracle Directory

Option	Description
Mandatory Attributes	Mandatory attributes of the object class for which you are searching. For example, the phrase <code>Mandatory Attributes Contains cn</code> gives you a list of all object classes in which the <code>cn</code> attribute is mandatory.
Optional Attributes	Optional attributes of the object class for which you are searching

Table A–23 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 <code>Type Begins With aux</code> 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 <code>Type Ends With ral</code> 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 <code>Optional Attributes Contains cn</code> gives you a list of all object classes in which <code>cn</code> 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 <code>Super Class Exact Match person</code> gives you a list of all object classes that have <code>person</code> 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 <code>Name Greater or Equal orcl</code> gives you a list of object classes from those beginning with the letters <code>orcl</code> to those beginning with letters at the end of the alphabet.
Less or Equal	Searches for an object class in which the property you selected is numerically or alphabetically less than or equal to the value you enter. For example, the phrase <code>Name Less or Equal orcl</code> gives you a list of object classes from those beginning with the letters <code>orcl</code> 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 <code>Mandatory Attributes Not Null</code> gives you a list of all object classes which contain mandatory attributes.

Table A–24 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 A–25 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.
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 A–26 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: <ul style="list-style-type: none"> ▪ 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
Super	Super attribute for each attribute

Table A–27 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 <code>Syntax Begins With 1.3</code> gives you a list of all attributes in which the first few numbers of the syntax identifier are <code>1.3</code> .
Ends With	Searches by using only the last few characters of the property's value. For example, the phrase <code>Name Ends With License</code> gives you a list of all attributes with that ending, such as <code>carLicense</code> .
Contains	Searches for attributes that include the property with the value you enter. For example, the phrase <code>Ordering Contains time</code> gives you a list of all attributes with the word <code>time</code> in the <code>Ordering</code> 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 <code>Equality Exact Match caseIgnoreMatch</code> gives you a list of all attributes that have the <code>caseIgnoreMatch</code> 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 <code>Name Greater or Equal orcl</code> gives you a list of attributes from those beginning with <code>orcl</code> 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 <code>Name Less or Equal orcl</code> gives you a list of attributes from those beginning with <code>orcl</code> 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 <code>Description Not Null</code> gives you a list of all attributes which have text in the description field.

Table A–28 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 A–29 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 .

Table A–29 (Cont.) Fields in the General Tab Page of the New Attribute Type Dialog

Field	Description
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 A–30 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.
Usage	Specify standards for how the attribute can be used. Options are: <ul style="list-style-type: none"> ▪ <code>userApplications</code> Attributes whose values must be entered by the user, for example, <code>telephoneNumber</code> ▪ <code>directoryOperation</code> Attributes whose values are entered by the directory server, for example, <code>creatorName</code> or <code>timeStamp</code> ▪ <code>distributedOperation</code> ▪ <code>dSAOperation</code> Attributes used for the internal operation of the server, for example, <code>orclUpdateSchedule</code>
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: <ol style="list-style-type: none"> 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. <p>To delete a super attribute from the Super field, select it, then choose Delete.</p>

Matching Rules Fields in Oracle Directory Manager

Table A–31 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 A–32 *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	<p>The auxiliary object classes whose attributes you want to associate with the specified structural object class. To specify an auxiliary class:</p> <ol style="list-style-type: none"> 1. Choose Add. The Auxiliary Class Selector dialog box appears. 2. Select the auxiliary class you want to add. 3. 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	<p>The mandatory attributes you want to associate with the specified structural object class. To specify a mandatory attribute:</p> <ol style="list-style-type: none"> 1. Choose Add. The Mandatory Attribute Selector dialog box appears. 2. Select the mandatory 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 mandatory attribute you just specified appears in the Mandatory Attributes field.
Optional Attributes	<p>The optional attributes you want to associate with the specified structural object class. To specify an optional attribute:</p> <ol style="list-style-type: none"> 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 A–33 *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

Table A–33 (Cont.) Fields in the Content Rule Dialog Box

Field	Description
Auxiliary Classes	<p>The auxiliary object classes whose attributes you want to associate with the specified structural object class. To specify an auxiliary class:</p> <ol style="list-style-type: none"> 1. Choose Add. The Auxiliary Class Selector dialog box appears. 2. Select the auxiliary class you want to add. 3. 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	<p>The mandatory attributes you want to associate with the specified structural object class. To specify a mandatory attribute:</p> <ol style="list-style-type: none"> 1. Choose Add. The Mandatory Attribute Selector dialog box appears. 2. Select the mandatory 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 mandatory attribute you just specified appears in the Mandatory Attributes field.
Optional Attributes	<p>The optional attributes you want to associate with the specified structural object class. To specify an optional attribute:</p> <ol style="list-style-type: none"> 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](#)
- [System Operational Attributes Fields in Oracle Directory Manager](#)
- [Super, Guest, and Proxy User Fields in Oracle Directory Manager](#)
- [Query Optimization Fields in Oracle Directory Manager](#)
- [Entry Search Fields and Buttons in Oracle Directory Manager](#)

Configuration Sets Fields in Oracle Directory Manager

See Also: [Managing Server Configuration Set Entries by Using Oracle Directory Manager](#) on page 5-3

Table A–34 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: <ul style="list-style-type: none"> ▪ RC4-56 ▪ DES ▪ 3DES

Table A–35 Fields in the Configuration Sets—SSL Settings Tab Page

Field	Description
SSL Authentication	Choose one of the following: <ul style="list-style-type: none"> ▪ 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: <ul style="list-style-type: none"> ▪ 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: <pre>file:/home/my_dir/my_wallet</pre> <p>On Microsoft Windows, you could set this parameter as follows:</p> <pre>file:C:\my_dir\my_wallet</pre>

Table A–35 (Cont.) Fields in the Configuration Sets—SSL Settings Tab Page

Field	Description
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 5-7

Table A–36 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: <code>ldap://host_name:port_number</code> See Also: "Setting the Alternate Server List by Using Oracle Directory Manager" in <i>Oracle Application Server High Availability Guide</i>	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 <code>orclStatsFlag</code> attribute also must be set to 1 for enabling this feature. See Also: "Configuring Critical Events" on page 10-16 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 A-36 (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 2-8, 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 ACL. 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
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

Table A–36 (Cont.) System Operation Attributes Displayed in Oracle Directory Manager

Field	Description	Default Value	Modifiable?
Password Encryption	Hash algorithm for encrypting the password. Options are: <ul style="list-style-type: none"> ▪ 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
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

Table A–36 (Cont.) System Operation Attributes Displayed in Oracle Directory Manager

Field	Description	Default Value	Modifiable?
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 <code>supportedcontrol</code> attribute in the root DSE. Each control type has an associated object identifier defined by the LDAP standard. The values of the <code>supportedcontrol</code> attribute are standard object identifiers assigned to control types.	<code>manageDSACtrl</code>	No
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: "Authentication by Using Simple Authentication and Security Layer (SASL)" on page 12-7	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 5-9

Table A–37 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 <code>orcladmin</code> .
Super User Password	Type the super user password. The default is the same as the password you specified for the Oracle Application Server administrator (<code>ias_admin</code>) during installation. You should change this password immediately.

Table A–37 (Cont.) Fields in the System Passwords Tab Page

Field	Description
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 <code>guest</code> .
Guest Login Password	Type the guest login password. The default is <code>guest</code> .
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 <code>proxy</code> .
Proxy Login Password	Type the proxy login password. The default is <code>proxy</code> . 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 21-9

Table A–38 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 21-9 for a discussion of skewed attributes
Common Name	The common name of the entry containing information about skewed attributes, namely, <code>dsainfo</code> . 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.
Object Class	The object classes associated with the <code>dsainfo</code> entry.
Time limit mode	When you set the server operation time limit as described in "Setting System Operational Attributes" on page 5-7, 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 A–39 Search Filters for Entries

Filter	Description
Begins With	Searches by using only the first few characters of the attribute's value. For example, <code>cn Begins With Fran</code> retrieves all entries in which the first few letters of the <code>cn</code> attribute are <code>Fran</code> . 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, <code>cn Ends With son</code> retrieves Baldissou, 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, <code>cn Contains Wins</code> retrieves all entries in which the <code>cn</code> attribute contains the letters <code>wins</code> . 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, <code>cn Exactly Matches Franklin Baldwins</code> retrieves all entries in which the <code>cn</code> attribute has the value <code>Franklin Baldwins</code> .
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, <code>cn Greater or Equal Frank</code> retrieves all entries with <code>cn</code> attributes that range from the first Frank to the end of the alphabet.
Less or Equal	Searches for entries in which the specified attribute is numerically or alphabetically less than or equal to the value you enter. For example, <code>cn Less or Equal Frank</code> retrieves all <code>cn</code> 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 <code>cn Present</code> retrieves all entries with the <code>cn</code> attribute at that level of the tree.

Table A–40 Buttons Used 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, <code>cn=Baldwins And title=Laborer</code> 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, <code>title=Laborer Or title=Foreman</code> 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, <code>cn=Frank And Not title=Laborer</code> retrieves all persons named Frank who are not laborers.
Delete	Deletes a selected search criteria bar

Table A–40 (Cont.) Buttons Used in Searches for Entries

Button	Description
Advanced	<p>Adds a search criteria bar when including attribute options in the search. Use this syntax: <i>attribute; attribute_option filter attribute_option_value</i></p> <p>For example, <i>cn; lang_sp=J*</i> retrieves all attribute option values for <i>cn; lang_sp</i> that begin with the letter J.</p> <p>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 <i>carLicense; lang_sp</i>, the <i>carLicense</i> attribute must be indexed before the <i>carLicense; lang_sp</i> attribute option can be used in searches.</p> <p>See Also:</p> <ul style="list-style-type: none"> ▪ "Indexing an Attribute by Using Oracle Directory Manager" on page 8-12 ▪ "Indexing an Attribute by Using Command-Line Tools" on page 8-14

SSL Management Fields in Oracle Directory Manager

See Also:

- [Table A–35, "Fields in the Configuration Sets—SSL Settings Tab Page"](#) on page A-21
- ["Configuring SSL Parameters by Using Oracle Directory Manager"](#) on page 13-3

Table A–41 Fields in the SSL Settings Tab Page

Field	Description
SSL Authentication	<p>Choose one of the following:</p> <ul style="list-style-type: none"> ▪ 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	<p>Choose one of the following:</p> <ul style="list-style-type: none"> ▪ 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	<p>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:</p> <pre>file:/home/my_dir/my_wallet</pre> <p>On Microsoft Windows, you could set this parameter as follows:</p> <pre>file:C:\my_dir\my_wallet</pre>
SSL Port	The default SSL port is 636. You can change the SSL port.

Synchronization Fields in Oracle Directory Manager

This section describes the fields in Oracle Directory Manager for administering directory synchronization. These are fields for registering a directory integration profile

Table A–42 *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 <code>MSAccess</code> creates an integration profile named <code>orclodipagentname=MSAccess,cn=subscriber profile,cn=changelog subscriber,cn=oracle internet directory</code> . 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 <code>IMPORT</code> .
Profile Status	Specify whether the profile is enabled or disabled. This field is mandatory. The default is <code>ENABLE</code> .
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 <code>welcome</code> .
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 <code>60</code> .
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 Oracle Directory Integration and Provisioning with which this profile was created.

Table A–43 *Fields on the Execution Tab for Synchronization in Oracle Directory Manager*

Field	Description
Agent Execution Command	<p>Specify the agent executable name and the arguments used by the directory integration and provisioning server to execute the agent.</p> <p>This field is optional. There is no default.</p> <p>A typical execution command is of the form,</p> <pre>odicmd user=%orclodipcondirAccessAccount pass=%orclodipcondiraccesspassword</pre> <p>Where <code>odicmd</code> is the command to be executed (available in the <code>PATH</code> or specified as a complete path name), and</p> <pre>user=%orclodipcondirAccessAccount pass=%orclodipcondiraccesspassword</pre> <p>are the command-line arguments. The value to be passed for the user is derived from the attribute <code>orclodipcondiraccessaccount</code>, and the value to be passed for <code>pass</code> is derived from the attribute <code>orclodipcondiraccesspassword</code>.</p> <p>A typical example is given in the Oracle Human Resources agent.</p>
Connected Directory Account	<p>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 <code>Scott</code>. If the connected directory is another LDAP-compliant directory, then the account might be <code>cn=Directory Manager</code>.</p> <p>This field is optional. There is no default.</p>
Connected Directory Account Password	<p>Specify the password the connector/agent is to use when accessing the connected directory. This field is optional. There is no default.</p>
Additional Config Info	<p>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 <code>ldapuploadagentfile.sh</code>. There is no default.</p>
Connected Directory URL	<p>Connect details required to connect to the connected directory. This parameter refers to the host name and port number as <code>host:port:sslmode</code></p> <p>To connect by using SSL, enter <code>host:port:1</code>.</p> <p>Make sure the certificate to connect to the directory is stored in the wallet, the location of which is specified in the file <code>odi.properties</code>.</p> <p>Note: To connect to SunONE Directory Server by using SSL, the server certificate needs to be loaded into the wallet.</p> <p>See Also: The chapter on Oracle Wallet Manager in <i>Oracle Advanced Security Administrator's Guide</i></p>
Interface Type	<p>The format used by the import or export file. Options are <code>DB</code>, <code>LDAP</code>, <code>LDIF</code>, and <code>TAGGED</code>. This field is optional. The default is <code>TAGGED</code>.</p>

Table A–44 *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 <code>dipassistant</code> . See the "dipassistant" command-line tool reference in <i>Oracle Identity Management User Reference</i> .
Connected Directory Matching Filter	Specify the attribute that uniquely identifies an entry in the connected directory.
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 A–45 *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.

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
Request for Comments: 2254
Category: Standards Track

T. Howes
Netscape Communications Corp.
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

Copyright (C) The Internet Society (1997). All Rights Reserved.

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

then 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

Copyright (C) The Internet Society (1997). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other

Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS 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.

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>)] [DenyGroupOverride] [AppendToAll]

<subject> ::= <entity> [<BindMode>] [Added_object_constraint=(<ldapFilter>)]
<entity> ::= * | self | dn="<regex>" | dnAttr=(<dn_attribute>) | group="<dn>" |
guidattr=(<guid_attribute>) | groupattr=(<group_attribute>) | [SuperUser]

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> )]+
```

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 2-15 prior to configuring Globalization Support

This chapter contains these topics:

- [About Character Sets and the Directory](#)
- [The NLS_LANG Environment Variable](#)
- [Using Non-AL32UTF8 Databases](#)
- [Using Globalization Support with LDIF Files](#)
- [Using Globalization Support with Command-Line Tools](#)
- [Setting NLS_LANG in the Client Environment](#)
- [Using Globalization Support with Bulk Tools](#)

About Character Sets and the Directory

When computer systems process characters, they use numeric codes instead of the graphical representation of the character. For example, when the database stores the letter A, it actually stores a numeric code that is interpreted by software as the letter.

A group of characters (for example, alphabetic characters, ideographs, symbols, punctuation marks, and control characters) can be encoded as a character set. Each encoded character set assigns a unique code to each character in the set. For example, in the ASCII encoding scheme, the character code of the first character of the English upper-case alphabet is 0x41; in the EBCDIC encoding scheme, it is 0xc1.

The computer industry uses many encoded character sets. These character sets can differ in the number and types of characters available and in many other ways as well.

When you create a database, you specify an encoded character set. Choosing a character set determines, among other things, what languages can be represented in the database.

Oracle supports most national, international, and vendor-specific encoded character set standards.

This section contains the following topics:

- [About Unicode](#)
- [About Oracle and UTF-8](#)
- [Migration from UTF8 to AL32UTF8 when Upgrading Oracle Internet Directory](#)

About Unicode

No single character set contains enough characters to meet the requirements of day-to-day e-business requirements. For example, no one national character set can represent all the languages in the European Union. Moreover, there are potential conflicts between character sets because the same character can be represented by different codes in different character sets.

To overcome these obstacles, a global character set, called Unicode, was developed. It is a universal encoded character set that can store information from any language including punctuation marks, diacritics, mathematical symbols, technical symbols, musical symbols, and so forth. As of version 3.2, the Unicode Standard supports over 95,000 characters from the world's alphabets, ideograph sets, and symbol collections. It includes 45,000 supplementary characters, most of which are Chinese, Japanese, and Korean characters that are rarely used but nevertheless need representation in electronic documentation.

Unicode has more than one implementation standard, and these are described in [Table D-1](#).

Table D-1 *Unicode Implementations*

Implementation	Description
UTF-8	A variable-width 8-bit encoding of Unicode. One Unicode character can be one, two, three, or four bytes. Characters from European scripts are represented in one or two bytes. Those from Asian scripts are represented in three bytes, and supplementary characters are represented in four.
UCS-2	A fixed-width 16-bit encoding of Unicode in which each character, regardless of the script, is two bytes.
UTF-16	The 16-bit encoding of Unicode. It is an extension of UCS-2 that supports the supplementary characters added in Unicode 3.1. One character can be two or four bytes. Characters from European and Asian scripts are represented in two bytes, and supplementary characters are represented in four.

About Oracle and UTF-8

Oracle began supporting Unicode as a database character set beginning with Oracle database version 7. With Oracle9i, Oracle added a new UTF-8 character set called AL32UTF8. This database character set supports the latest version of Unicode (3.2), including the latest supplementary characters. Oracle intends to enhance AL32UTF8 as necessary to support future versions of the Unicode standard.

Migration from UTF8 to AL32UTF8 when Upgrading Oracle Internet Directory

Oracle Internet Directory now supports AL32UTF8. If you have upgraded Oracle Internet Directory from a version prior to 10g Release 2 (10.1.2), then, for better

performance, Oracle recommends that you change the character set for the directory database from UTF8 to AL32UTF8. To do this:

1. Run the character set scanner (CSSCAN) to ensure that there are no invalid UTF8 characters inside your current database.
2. Run the CSALTER script to update the database to AL32UTF8.

See Also: The chapter on character set migration in the *Oracle Database Globalization Support Guide* in the Oracle Database Documentation Library

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.

Table D-2 Components of the NLS_LANG Parameter

Component	Description
language	<p>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.</p> <p>If language is not specified, the value defaults to American English.</p> <p>See Also: <i>Oracle Database Globalization Support Guide</i> in the Oracle Database Documentation Library for a complete list of languages</p>
territory	<p>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.</p> <p>If territory is not specified, the value defaults to America.</p> <p>See Also: <i>Oracle Database Globalization Support Guide</i> in the Oracle Database Documentation Library for a complete list of territories</p>
charset	<p>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, WE8MSWIN1252, JA16SJIS, or AL32UTF8.</p> <p>See Also: <i>Oracle Database Globalization Support Guide</i> in the Oracle Database Documentation Library for a complete list of character sets</p>

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-AL32UTF8 Databases

You can run the Oracle directory server and database tools on a non-AL32UTF8 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: "LDIF file formatting rules and examples" in *Oracle Identity Management User Reference*

Attribute types are always ASCII strings that cannot contain multibyte characters. Oracle Internet Directory does not support multibyte characters in attribute type names. However, Oracle Internet Directory does support attribute *values* containing multibyte characters such as those in the simplified Chinese (ZHS16GBK) character set.

Attribute values can be encoded in different ways to allow Oracle Internet Directory tools to interpret them properly. There are two scenarios:

- [An LDIF file Containing Only ASCII Strings](#)
- [An LDIF file Containing UTF-8 Encoded Strings](#)

An LDIF file Containing Only ASCII Strings

In this scenario, character strings for attribute values are also in ASCII.

Because all tools use the UTF-8 character set by default, and ASCII is a proper subset of UTF-8, all tools can interpret these files. The same is true of keyboard input of values that are simply ASCII strings.

An LDIF file Containing UTF-8 Encoded Strings

In this scenario, character strings for attribute values are also in UTF-8.

Because, by default, all tools use the UTF-8 character set, all tools can interpret these files. The same is true of keyboard input of values that 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 character 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](#)
- [Examples: Using the -E Argument with Command-Line Tools](#)

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 GBK to Unicode UTF-8 before being sent to the directory 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 AL32UTF8 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"` before they are sent across the wire to the directory server.

Examples: Using the `-E` Argument with Command-Line Tools

[Table D-3](#) 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 AL32UTF8. For example, in each command, the values for the `-D` and `-w` options are in GBK. Specifying the `-E` argument converts them to UTF-8.

Note that, in the examples in [Table D-3](#), we do not show any actual characters belonging to the `.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: "Oracle Internet Directory Server Administration Tools" in *Oracle Identity Management User Reference* for syntax and usage notes for each of the command-line tools

Table D-3 Examples: Using the `-E` Argument with Command-Line Tools

Tool	Example
<code>ldapbind</code>	<code>ldapbind -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password</code>
<code>ldapsearch</code>	<code>ldapsearch -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password</code>
<code>ldapadd</code>	<code>ldapadd -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password</code>
<code>ldapaddmt</code>	<code>ldapaddmt -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password</code>
<code>ldapmodify</code>	<code>ldapmodify -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password</code>
<code>ldapmodifymt</code>	<code>ldapmodifymt -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password</code>
<code>ldapdelete</code>	<code>ldapdelete -h my_host -p 389 -E ".ZHS16GBK" -D "o=acme,c=us" -w my_password</code>

Table D-3 (Cont.) Examples: Using the -E Argument with Command-Line Tools

Tool	Example
ldapcompare	<pre>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</pre>
ldapmoddn	<pre>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</pre>

Setting NLS_LANG in the Client Environment

If the output required by the client is UTF-8, then you do not need to set the NLS_LANG environment variable. In this case, the character set component of 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 AL32UTF8 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 AL32UTF8 character set.

Note: If you are using Microsoft Windows, 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. AL32UTF8 cannot be used. See the *Oracle Database Installation Guide for Microsoft Windows (32-Bit)* 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 Oracle Database with Oracle Internet Directory, then you must also set the database character set to AL32UTF8.

See Also: *Oracle Database Globalization Support Guide* in the Oracle Database Documentation Library and *Oracle Database Installation Guide for Microsoft Windows (32-Bit)*

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 Idifwrite](#)
- [Using Globalization Support with bulkdelete](#)
- [Using Globalization Support with bulkmodify](#)

See Also: "Oracle Internet Directory Server Administration Tools" in *Oracle Identity Management User Reference* 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 8.6 or later. Visit:
<http://www.datafocus.com/>
-
-

Using Globalization Support with Idifwrite

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 `AL32UTF8`—for example, `".ZHS16GBK"`—then the output LDIF file will contain a BASE64 encoded value of simplified Chinese GBK 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 GBK, 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 8.6 or later. 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 GBK 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 8.6 or later. Visit: <http://www.datafocus.com/>
-
-

Setting up Access Controls for Creation and Search Bases for Users and Groups

If you modify the User Search Base, the User Creation Base, the Group Search Base, or the Group Creation Base, then access controls for the new container need to be set up properly. This appendix contains these topics:

- [Setting up Access Controls for the User Search Base and the User Creation Base](#)
- [Setting up Access Controls for the Group Search Base and the Group Creation Base](#)

Setting up Access Controls for the User Search Base and the User Creation Base

To set up access controls for the User Search Base and the User Creation Base:

1. Create an LDIF (`user_aci.ldif`) file with the following entry:

```
--- BEGIN LDIF file contents---
dn: %usersearch_or_createbase_dn%
changetype: modify
add: orclaci
orclaci: access to entry by group="cn=oracledascreateuser,
cn=groups,cn=OracleContext,%subscriberdn%"
added_object_constraint=(objectclass=orcluser*) (browse,add) by
group="cn=Common User Attributes, cn=Groups,
cn=OracleContext,%subscriberdn%" (browse) by
group="cn=PKIAdmins, cn=groups, cn=OracleContext,%subscriberdn%" (browse)
orclaci: access to entry filter=(objectclass=inetorgperson) by
group="cn=oracledascreateuser, cn=groups,cn=OracleContext,%subscriberdn%"
added_object_constraint=(objectclass=orcluser*) (browse,add) by
group="cn=oracledasdeleteuser, cn=groups,cn=OracleContext,%subscriberdn%"
(browse,delete) by group="cn=oracledasedituser,
cn=groups,cn=OracleContext,%subscriberdn%" (browse) by
group="cn=UserProxyPrivilege, cn=Groups,cn=OracleContext,%subscriberdn%"
(browse,
proxy) by dn="orclApplicationCommonName=DASApp, cn=DAS,
cn=Products,cn=oraclecontext" (browse,proxy) by self (browse, nodelete, noadd)
by
group="cn=Common User Attributes, cn=Groups,cn=OracleContext,%subscriberdn%"
(browse) by * (browse, noadd, nodelete)
orclaci: access to attr=(*) filter=(objectclass=inetorgperson) by
group="cn=oracledasedituser, cn=groups,cn=OracleContext,
%subscriberdn%" (read,search,write,compare) by self (
read,search,write,selfwrite,compare) by *
(read, nowrite, nocompare)
```

```

orclaci: access to attr=(userPassword)
  filter=(objectclass=inetorgperson) by
  group="cn=OracleUserSecurityAdmins,cn=Groups,
  cn=OracleContext,%subscriberdn%"
  (read,search,write,compare) by group="cn=oracledasedituser,
  cn=groups,cn=OracleContext,%subscriberdn%"
  (read,search,write,compare) by self
  (read,search,write,selfwrite,compare) by group="cn=authenticationServices,
  cn=Groups,cn=OracleContext,%subscriberdn%" (compare) by * (none)
orclaci: access to attr=(authpassword, orclpasswordverifier, orclpassword) by
  group="cn=oracledasedituser,cn=groups,cn=OracleContext,%subscriberdn%"
  (read,search,write,compare) by
  group="cn=verifierServices,cn=Groups,cn=OracleContext,%subscriberdn%"
  (search, read, compare) by self (search,read,write,compare) by * (none)
orclaci: access to attr=(orclpwdaccountunlock) by
  group="cn=oracledasedituser,cn=groups,cn=OracleContext,%subscriberdn%" (
  write) by * (none)
orclaci: access to attr=(usercertificate, usersmimecertificate) by
  group="cn=PKIAdmins,cn=Groups,cn=OracleContext,%subscriberdn%"
  (read, search, write, compare) by self (read, search, compare) by *
  (read, search, compare)
orclaci: access to attr=(mail) by
  group="cn=EmailAdminsGroup,cn=EmailServerContainer,cn=Products,
  cn=OracleContext" (write) by group="cn=oracledasedituser,
  cn=groups,cn=OracleContext,%subscriberdn%" (read,search,write,compare)
orclaci: access to attr=(orclguid, orclisenabled, modifytimestamp,mail)
  by group="cn=Common User Attributes,
  cn=Groups,cn=OracleContext,%subscriberdn%"
  (read, search, compare) by group="cn=oracledasedituser,
  cn=groups,cn=OracleContext,%subscriberdn%" (read,search,write,compare)
  by * (read, nowrite, nocompare)
orclaci: access to attr=(orclpasswordhintanswer) by
  group="cn=Common User Attributes,
  cn=Groups,cn=OracleContext,%subscriberdn%" (read, search, compare) by self
  (read,search,write,selfwrite,compare) by * (noread, nowrite, nocompare)
orclaci: access to attr=(orclpasswordhint) by
  group="cn=Common User Attributes,
  cn=Groups,cn=OracleContext,%subscriberdn%" (read, search, compare) by self
  (read,search,write,selfwrite,compare) by
  group="cn=OracleUserSecurityAdmins,cn=Groups,cn=OracleContext,
  %subscriberdn%" (read,search,write,compare) by *
  (noread, nowrite, nocompare)
orclaci: access to attr=(displayName, preferredlanguage,
  orcltimezone,orcldateofbirth,orclgender,orclwirelessaccountnumber,cn,
  uid,homephone,telephonenumber) by group="cn=Common User Attributes,
  cn=Groups,cn=OracleContext,%subscriberdn%"
  (read, search, compare) by group="cn=oracledasedituser,
  cn=groups,cn=OracleContext,%subscriberdn%" (read,search,write,compare)
  by self (read,search,write,selfwrite,compare) by *
  (read, nowrite, nocompare)
-
add: orclentrylevelaci
orclentrylevelaci: access to entry by group="cn=oracledascreateuser,
  cn=groups,cn=OracleContext,%subscriberdn%" added_object_constraint=
  (objectclass=orcluser*) (browse, add) by * (browse)
---END LDIF file contents-----

```

2. Replace `%subscriberdn%` with the dn of the subscriber and `%usersearch_or_createbase_dn%` with the new value of the container DN where the new user search/create base points to.

3. Run the ldapmodify command as follows:

```
ldapmodify -p oidport -h oidhost -D cn=orcladmin -w Instance Password -v \
-f user_aci.ldif
```

Setting up Access Controls for the Group Search Base and the Group Creation Base

To set up access controls for the Group Search Base and the Group Creation Base:

1. Create an ldif (group_aci.ldif) file with the following entry:

```
--- BEGIN LDIF file contents---
dn: %groupsearch_or_createbase_dn%
changetype: modify
add: orclaci
orclaci: access to entry by group="cn=IASAdmins,
cn=groups,cn=OracleContext,%subscriberdn%"
added_object_constraint=(objectclass=orclcontainer) (browse,add)
orclaci: access to entry by group="cn=oracledascreategroup,
cn=groups,cn=OracleContext,%subscriberdn%"
added_object_constraint=(objectclass=orclgroup*) (browse,add) by
group="cn=Common
Group Attributes, cn=Groups,cn=OracleContext,%subscriberdn%" (browse)
orclaci: access to entry filter=(!(objectclass=orclgroup)(orclisvisible=false))
by
groupattr=(owner) (browse, add, delete) by dnattr=(owner)
(browse, add, delete) by
group="cn=Common Group Attributes, cn=Groups,cn=OracleContext,%subscriberdn%"
(browse) by * (none)
orclaci: access to entry
filter=(!(objectclass=orclgroup)(!(orclisvisible=false))) by
group="cn=oracledascreategroup, cn=groups,cn=OracleContext,%subscriberdn%"
added_object_constraint=(objectclass=orclgroup) (browse,add) by
group="cn=oracledasdeletigroup, cn=groups,cn=OracleContext,%subscriberdn%"
(browse,delete) by group="cn=oracledaseditgroup,
cn=Groups,cn=OracleContext,%subscriberdn%" (browse) by groupattr=(owner) (
browse,
add, delete) by dnattr=(owner) (browse, add, delete) by group="cn=Common Group
Attributes, cn=Groups,cn=OracleContext,%subscriberdn%" (browse)
orclaci: access to attr=(*)
filter=(!(objectclass=orclgroup)(orclisvisible=false)) by
groupattr=(owner) (read,search,write,compare) by dnattr=(owner)
(read,search,write,compare) by * (none) by group="cn=Common Group Attributes,
cn=Groups,cn=OracleContext,%subscriberdn%" (read, search, compare)
orclaci: access to attr=(*)
filter=(!(objectclass=orclgroup)(!(orclisvisible=false))) by
groupattr=(owner) (read,search,write,compare) by dnattr=(owner)
(read,search,write,compare) by group="cn=oracledaseditgroup,
cn=groups,cn=OracleContext,%subscriberdn%" (read,search,write,compare) by
group="cn=Common Group Attributes, cn=Groups,cn=OracleContext,%subscriberdn%"
(read, search, compare)
-
add: orclentrylevelaci
orclentrylevelaci: access to entry by group="cn=oracledascreategroup,
cn=groups,cn=OracleContext,%subscriberdn%"
added_object_constraint=(objectclass=orclgroup) (browse, add) by
group="cn=IASAdmins, cn=groups,cn=OracleContext,%subscriberdn%"
added_object_constraint=(objectclass=orclcontainer) (browse,add) by * (browse)
---END LDIF file contents-----
```

2. Replace `%subscriberdn%` with the DN of the subscriber and `%groupsearch_or_createbase_dn%` with the new value of the container DN where the new group search base or group create base points to.
3. Run the `ldapmodify` command as follows:

```
ldapmodify -p oidport -h oidhost -D cn=orcladmin -w instance password \  
-v -f group_aci.ldif
```

The Multimaster Replication Process

This appendix describes how the multimaster replication process adds, deletes, and modifies entries, and how it modifies DNs and RDNs. It contains these topics:

- [How the Multimaster Replication Process Adds a New Entry to a Consumer](#)
- [How the Multimaster Replication Process Deletes an Entry](#)
- [How the Multimaster Replication Process Modifies an Entry](#)
- [How the Multimaster Replication Process Modifies a Relative Distinguished Name](#)
- [How the Multimaster Replication Process Modifies a Distinguished Name](#)

How the Multimaster Replication Process Adds a New Entry to a Consumer

When the directory replication server 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. You 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.

Searching the Directory for User Certificates

Starting with 10g Release 2 (10.1.2), you can perform a command-line search of the binary attribute `usercertificate`.

Prior to 10g Release 2 (10.1.2.0.2), the only way to identify a user from the certificate was through the DN specified in the certificate. This is known as certificate matching. Starting with 10g Release 2 (10.1.2.0.2), Oracle Internet Directory supports certificate mapping, in addition to certificate matching. Certificate matching requires that a user certificate be provisioned in the directory. Certificate mapping does not require provisioning of a user certificate.

This chapter includes the following topics:

- [Certificate Mapping](#)
- [Search Types](#)

Certificate Mapping

Certificate mapping allows a customer to define rules for mapping the certificate to the user's DN. A certificate mapping rule is a set of rules for parsing the certificate and for querying the directory for the user's identity. Only custom extensions of certificates can be used in mapping rules.

The following examples show how to add, delete, and modify a certificate mapping rule.

Adding a Certificate Mapping Rule

Add a mapping rule using `ldapmodify`, as follows:

```
ldapmodify -h hostName -p port_number -f certMapRuleAdd.ldif
```

The file `certMapRuleAdd.ldif` should look something like this:

```
dn: cn=maprule1,cn=SASL-EXTERNAL,cn=Identity Mapping Configurations,cn=Server  
Configurations  
cn: maprule1  
objectclass: orclidmapping  
objectclass: orclcertidmapping  
orclSearchScope: subtree  
orclSearchFilter: (cn=${(2.16.750.5.14.2.81.2.5.1)})  
orclcertExtensionOID: 2.16.750.5.14.2.81.2.5  
orclcertExtensionAttribute: 2.16.750.5.14.2.81.2.5.1
```

Deleting a Certificate Mapping Rule

Delete a mapping rule using `ldapdelete`, as follows:

```
ldapdelete hostName -p port_number "cn=maprule1,cn=SASL-EXTERNAL,cn=Identity
Mapping Configurations,cn=Server Configurations"
```

Modifying a Certificate Mapping Rule

Modify a mapping rule using `ldapmodify`, as follows:

```
ldapmodify -h hostName -p port_number -f certMapRuleMod.ldif
```

The file `certMapRuleMod.ldif` should look something like this:

```
dn: cn=maprule1,cn=SASL-EXTERNAL,cn=Identity Mapping Configurations,cn=Server
Configurations
changetype:modify
replace: <attrName>
<attrName>: <attrValue>
```

Search Types

You can use two kinds of `ldapsearch` filters:

- A filter of the form `"usercertificate=certificate_serial_number$certificate_issuer_DN"`. A combination of the certificate serial number and the certificate issuer's DN is used to locate the certificate. This combination is called the certificate match value.
- A filter of the form `"usercertificate;binary=base_64_encoded_value_of_certificate"`. Using this filter, one of six types of searches is possible, depending upon two things:
 - The value of the DSA configuration set attribute (DN: `"cn=dsaconfig,cn=configsets,cn=oracle internet directory"`), `orclpkimatchingrule`.
 - The presence or absence of the LDAP control `2.16.840.1.113894.1.8.23`

The six types of searches possible with a filter of the form

`"usercertificate;binary=base_64_encoded_value_of_certificate"` are:

Presence of LDAP control	Value of <code>orclpkimatchingrule</code>	Search Behavior
Absent	Not used	The hashed value of the client certificate is used to locate <code>usercertificate</code> .
Present	0	An exact-match search is performed. The subject DN of the client certificate is the search base. This DN is compared with the user DN in the directory. The search scope is Base. The filter is <code>"objectclass=*"</code> .
Present	1	The hashed value of the client certificate is used to locate <code>usercertificate</code> .
Present	2 (Default)	The hashed value of the client certificate is used to locate <code>usercertificate</code> . If this search yields nothing, An exact-match search is performed.

Presence of LDAP control	Value of orclpkimatchingrule	Search Behavior
Present	3	The mapping rule is used.
Present	4	First, the mapping rule is used. If that search yields nothing, then the search proceeds as if the value were 2.

Use the `ldapmodify` tool to set `orclpkimatchingrule` to the desired value.

Notes:

- The `usercertificate` attribute cannot be searched using a substring filter.
 - In an exact-match search, the search filter can contain only one attribute value assertion.
 - Only one-level and subtree searches are supported.
 - The `catalog.sh` tool does not support catalogs for user certificates—namely `ct_orclcertificatehash` and `ct_orclcertificatematch`.
 - The introduction in 10g Release 2 (10.1.2) of certificate hash values requires that certificates be upgraded from earlier releases. See the "upgradecert.pl" command-line tool reference in *Oracle Identity Management User Reference*.
-
-

See Also: ["Direct Authentication"](#) on page 12-3

LDAP Replica States

This chapter describes the replica states for LDAP-based replication. LDAP replica states have no effect upon Advanced Replication.

When LDAP based replication is configured, and the replication server is started, the server reads the replica state `orclreplicastate` from the local replica, "`orclreplicaid=local_Replica_ID, cn=replication configuration`". The replication server behaves differently, based upon the local replica state, as shown in [Table H-1](#). The replication server reads the local replica state from the local (consumer) node.

Table H-1 LDAP Replica States

Value	Meaning	Replication Server Behavior
0	Bootstrap	<p>Starts replication bootstrap processing to synchronize the consumer directory from the supplier, based on the replication naming context configuration. Updates the replica state to correspond with bootstrap progress.</p> <ul style="list-style-type: none"> ▪ Sets the replica state to 3 (Bootstrap in progress) immediately when it starts to bootstrap. ▪ Sets the replica state to 4 (Bootstrap in progress, <code>cn=oraclecontext</code> bootstrap has completed) after it completes bootstrapping of "<code>cn=oraclecontext</code>" successfully ▪ Sets the replica state to 5 (Bootstrap Error occurred) after bootstrap is completed but failure(s) detected during the bootstrap. Then it waits until the replica state is re-set. Note: Human intervention is required; See "Troubleshooting Oracle Internet Directory Replication" on page J-15 for details. ▪ Sets the replica state to 1 (On-line) after bootstrap has completed successfully. Then replication will automatically start to perform normal replication processing.
1	On line	Starts normal replication processing to replicate changes from the supplier to the consumer.
2	Off line	<p>Logs an error message in <code>oidrepld.log</code> similar to this:</p> <pre>2004/09/24:17:41:44 * Replica(dlsun1418_replica2) is in OFFLINE mode, Please update the replica state and restart OIDREPLD...</pre> <p>The administrator must set the replica state properly and restart the replication server.</p>
3	Bootstrap in progress	Sets the replica state back to 0 (Bootstrap), then starts to bootstrap again as if the replica state were 0.

Table H-1 (Cont.) LDAP Replica States

Value	Meaning	Replication Server Behavior
4	Bootstrap in progress, cn=oraclecontext bootstrap has completed.	Sets the replica state back to 0 (Bootstrap), then starts to bootstrap again as if the replica state were 0.
5	Bootstrap completed; failure detected for one or more naming contexts.	Logs an error message in oidrepld.log similar to this: 2004/09/24:17:13:30 * Replication BOOTSTRAP_ERROR mode detected for replica(dlsun1418_replica2) Then it waits until the replica state is reset properly.
6	Database copy-Based addnode; used only in Oracle Database Advanced Replication-based replication	This mode indicates that the replica is a database copy-based addnode for Oracle Database Advanced Replication-based replication. This replica state is not to be used in LDAP-based replication.

Oracle Internet Directory replication server logs the bootstrapping process in the Oracle Internet Directory replication server log, \$ORACLE_HOME/ldap/log/oidrepld00.log.

If bootstrap completes successfully, the log looks similar to the following example, and the replication server will automatically start to perform normal replication processing.

```
2004/10/06:17:13:25 * Starting OIDREPLD against isunnad03:5555...
2004/10/06:17:13:26 * Starting scheduler...
2004/10/06:17:13:27 * Start to BootStrap from supplier=isunnad03_purify to
consumer=isunnad03_purify3
2004/10/06:17:13:28 * gslrbssSyncDIT:Replicating namingcontext=cn=oraclecontext
.....
2004/10/06:17:14:21 * gslrbssSyncDIT:Sync done successfully for namingctx:
cn=oraclecontext, 222 entries matched
2004/10/06:17:14:21 * gslrbssSyncDIT:Replicating namingcontext=c=india .....
2004/10/06:17:14:21 * gslrbssSyncDIT:Sync done successfully for namingctx:
c=india, 0 entries matched
2004/10/06:17:14:21 * gslrbssSyncDIT:Replicating namingcontext=c=uk .....
2004/10/06:17:19:57 * gslrbssSyncDIT:Sync done successfully for namingctx: c=uk,
1087 entries matched
2004/10/06:17:19:57 * gslrbssSyncDIT:Replicating
namingcontext=cn=oracleschemaversion .....
2004/10/06:17:19:59 * gslrbssSyncDIT:Sync done successfully for namingctx:
cn=oracleschemaversion, 10 entries matched
2004/10/06:17:20:01 * gslrbsbBootStrap: BOOTSTRAP DONE SUCCESSFULL
```

If failures are detected, the log looks similar to the following example:

```
2004/09/14:12:57:23 * Starting OIDREPLD against dlsun1418:4444...
2004/09/14:12:57:25 * Starting scheduler...
2004/09/14:12:57:26 * Start to BootStrap from supplier=dlsun1418_replica to
consumer=dlsun1418_replica2
2004/09/14:12:57:27 * gslrbssSyncDIT:Replicating namingcontext=cn=oraclecontext
.....
```

```
2004/09/14:12:58:21 * gslrbssSyncDIT:Sync done successfully for namingctx:
cn=oraclecontext, 222 entries matched
2004/09/14:12:58:21 * gslrbssSyncDIT:Replicating namingcontext=cn=quan zhou .....
2004/09/14:12:58:23 * BootStrap failure when adding DN=cn=Quan
Zhou,server=dlsun1418_replica2,err=Constraint violation.
2004/09/14:12:58:23 * gslrbssSyncDIT:Sync failed for namingctx: cn=quan zhou, only
1 entries retrieved
2004/09/14:12:58:23 * gslrbssSyncDIT:Replicating
namingcontext=cn=oracleschemaversion .....
2004/09/14:12:58:25 * gslrbssSyncDIT:Sync done successfully for namingctx:
cn=oracleschemaversion, 10 entries matched
2004/09/14:12:58:51 * gslrbsbBootStrap: Failure occured when bootstrapping 1 out
of 3 namingcontext(s) from the supplier
```

Tip: You have two options for troubleshooting bootstrap failure.

- Option 1: Identify the cause of the bootstrap failure and fix the cause, then restart bootstrapping by setting the consumer replica's `orclreplicastate` to `Bootstrap` mode.
- Option 2: Identify the naming contexts that failed to be bootstrapped and use `oidreconcile` to reconcile them. Then resume replication by setting the consumer replica's `orclreplicastate` to `Online` mode.

Note: `Oidrepld` is now in `Bootstrap_error` mode, so you will need to reset the consumer replica's replica state (`orclreplicastate`).

Addition of a Directory Node by Using the Database Copy Procedure

This appendix explains how to add a new node to an existing replicating system by using the database copy procedure, also known as cold backup. This procedure works only for Oracle Internet Directory. Do not employ this procedure if other Oracle Identity Management components, such as Oracle Application Server Certificate Authority or Oracle Application Server Single Sign-On, are installed. You can use the database copy procedure, to create a new Directory Replication Group if you have a standalone Oracle Internet Directory node. This procedure is applicable only for Oracle Database Advanced Replication-based directory replication.

This chapter contains these topics:

- [Definitions](#)
- [Prerequisites](#)
- [Sponsor Directory Site Environment](#)
- [New Directory Site Environment](#)
- [Tasks To Be Performed on the Sponsor Node](#)
- [Tasks To Be Performed on the New Node](#)
- [Verification Process](#)

Definitions

The sponsor site refers to the site or host or node where Oracle Internet Directory and its repository, the Oracle database, are installed. The sponsor site is also referred to as sponsor node.

The new site refers to the site or host or node to which you are copying the Oracle Internet Directory repository. The new site is also referred to as new node.

Prerequisites

Your computing environment must meet the following prerequisites before you start this procedure:

1. The operating system, version, and patch level of the new directory site must be the same as that of the sponsor directory site. This procedure might not work if the patch level of the operating system differs.
2. Oracle Corporation strongly recommends that you back up the sponsor directory's repository before you employ this procedure.

3. Because this procedure involves copying Oracle data files, performance depends on the underlying network. If the underlying network is slow, it might be more efficient to use the method described in [Chapter 25, "Oracle Internet Directory Replication Administration"](#) to set up a replication group. Alternatively, you could physically transfer compressed Oracle data files on removable media. Consult your local system or network administrator for information about the network.
4. Only a person familiar with the Oracle database should implement this procedure.

Sponsor Directory Site Environment

In the example shown throughout this chapter, the sponsor directory site's environment is as follows:

```
Hostname = rst-sun
Domain name = acme.com
ORACLE_BASE = /private/oracle/app/oracle
ORACLE_HOME = /private/oracle/app/oracle/product/OraHome_1
ORACLE_SID = LDAP
LD_LIBRARY_PATH = $ORACLE_HOME/lib
NLS_LANG = AMERICAN_AMERICA.AL32UTF8
datafile location = /private/oracle/oradata/LDAP
Dump destination = /private/oracle/app/oracle/admin/LDAP/pfile,
                  /private/oracle/app/oracle/admin/LDAP/bdump,
                  /private/oracle/app/oracle/admin/LDAP/cdump,
                  /private/oracle/app/oracle/admin/LDAP/udump,
                  /private/oracle/app/oracle/admin/LDAP/create
```

New Directory Site Environment

In the example shown throughout this chapter, the new directory site's environment is as follows:

```
Hostname = dsm-sun
Domain name = acme.com
ORACLE_BASE = /privatel/oracle/app/oracle
ORACLE_HOME = /privatel/oracle/app/oracle/product/OraHome_1
ORACLE_SID = NLDAP
LD_LIBRARY_PATH = $ORACLE_HOME/lib
NLS_LANG = AMERICAN_AMERICA.UTF8
datafile location = /privatel/oracle/oradata/NLDAP
Dump destination = /privatel/oracle/app/oracle/admin/NLDAP/pfile,
                  /privatel/oracle/app/oracle/admin/NLDAP/bdump,
                  /privatel/oracle/app/oracle/admin/NLDAP/cdump,
                  /privatel/oracle/app/oracle/admin/NLDAP/udump,
                  /privatel/oracle/app/oracle/admin/NLDAP/create
```

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. Install Identity Management and Metadata Repository using Oracle Universal Installer. You need to install only Oracle Internet Directory, so select only Oracle Internet Directory in the **Select Configuration Options** screen of the installer. During the installation process, set ORACLE_SID to NLDAP and global name to NLDAP.ACME.COM.

See Also: "Installing OracleAS Infrastructure" in *Oracle Application Server Installation Guide*.

3. Issue the following command to copy all install-specific entries to sponsor directory:

```
remtool -backupmetadata \
  -master "sponsor_host:master_port/sponsor_repl_dn_pwd" \
  -replica "new_node_host:new_node_port/new_node_repldn_pwd"
```

where *sponsor_host*, *sponsor_port*, and *sponsor_repldn_pwd* are the hostname, port number, and replication DN password for the sponsor node, respectively.

See Also: "If an Existing Master is Used as a Remote Master Site" on page 25-6 in Chapter 25, "Oracle Internet Directory Replication Administration"

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 RESETLOGS;
```

This command creates a trace file under the user dump destination directory. In this example, that directory is

```
/private/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. Shut down 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
```

You can also stop OIDMON and the LDAP server process using `opmn`. Ensure that the replication server, directory server and OIDMON are shut down before you proceed to next step.

In these commands, *connect_string* is the net service name in the node's `tnsnames.ora` file.

3. On the other nodes of the directory replication group (DRG), shut down 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. Perform this step only if adding a node to an existing DRG. Quiesce Oracle Database Advanced Replication by running the following commands at the master definition site (MDS):

```
cd $ORACLE_HOME/ldap/admin
```

At the command line prompt execute SQL*Plus.

```
$ sqlplus /nolog
SQL> connect repadmin/repadmin_password;
SQL> @oidrsusp.sql
```

Note: Perform this procedure only on the master definition site.

At this point, other nodes are available for LDAP edits only, but replication will not take place.

5. Shut down the database and Oracle Net Services listener on the sponsor node only:

```
$ lsnrctl [listener_name] stop
$ sqlplus /nolog
SQL> connect /as sysdba
SQL> shutdown normal
SQL> exit
```

By default, the listener name is LISTENER.

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 all the lines except the `STARTUP NOMOUNT` and `CREATE CONTROLFILE` statements. After editing, `newdb.sql` should look like this:

```
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "LDAP" RESETLOGS NOARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 100
    MAXINSTANCES 8
    MAXLOGHISTORY 454
LOGFILE
  GROUP 1 '/private/oracle/oradata/LDAP/redo01.log' SIZE 10M,
  GROUP 2 '/private/oracle/oradata/LDAP/redo02.log' SIZE 10M,
  GROUP 3 '/private/oracle/oradata/LDAP/redo03.log' SIZE 10M
-- STANDBY LOGFILE
DATAFILE
  '/private/oracle/oradata/LDAP/system01.dbf',
  '/private/oracle/oradata/LDAP/sysaux01.dbf',
  '/private/oracle/oradata/LDAP/users01.dbf',
  '/private/oracle/oradata/LDAP/dcm.dbf',
  '/private/oracle/oradata/LDAP/portal.dbf',
  '/private/oracle/oradata/LDAP/ptldoc.dbf',
  '/private/oracle/oradata/LDAP/ptlidx.dbf',
  '/private/oracle/oradata/LDAP/ptllog.dbf',
  '/private/oracle/oradata/LDAP/oca.dbf',
```

```

'/private/oracle/oradata/LDAP/discopltml.dbf',
'/private/oracle/oradata/LDAP/discopltml.dbf',
'/private/oracle/oradata/LDAP/oss_sys01.dbf',
'/private/oracle/oradata/LDAP/wcrsys01.dbf',
'/private/oracle/oradata/LDAP/uddisys01.dbf',
'/private/oracle/oradata/LDAP/b2b_dt.dbf',
'/private/oracle/oradata/LDAP/b2b_rt.dbf',
'/private/oracle/oradata/LDAP/b2b_idx.dbf',
'/private/oracle/oradata/LDAP/b2b_lob.dbf',
'/private/oracle/oradata/LDAP/bam.dbf',
'/private/oracle/oradata/LDAP/orabpel.dbf',
'/private/oracle/oradata/LDAP/attrs1_oid.dbf',
'/private/oracle/oradata/LDAP/battrs1_oid.dbf',
'/private/oracle/oradata/LDAP/gcats1_oid.dbf',
'/private/oracle/oradata/LDAP/gdefault1_oid.dbf',
'/private/oracle/oradata/LDAP/svrmgl_oid.dbf',
'/private/oracle/oradata/LDAP/ias_meta01.dbf',
'/private/oracle/oradata/LDAP/undotbs.dbf'
CHARACTER SET AL32UTF8
;

```

8. Make following changes to newdb.sql.

a. Change the line:

```
CREATE CONTROLFILE REUSE DATABASE "LDAP" RESETLOGS NOARCHIVELOG
```

to say:

```
CREATE CONTROLFILE REUSE SET DATABASE "NLDAP" RESETLOGS NOARCHIVELOG
```

b. Modify the UNIX directory location of the database and logfiles to point to the new node site's directory.

In our example, after these modifications, newdb.sql should look like this:

```

STARTUP NOMOUNT
CREATE CONTROLFILE REUSE SET DATABASE "NLDAP" RESETLOGS NOARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 100
    MAXINSTANCES 8
    MAXLOGHISTORY 454
LOGFILE
    GROUP 1 '/private1/oracle/oradata/NLDAP/redo01.log' SIZE 10M,
    GROUP 2 '/private1/oracle/oradata/NLDAP/redo02.log' SIZE 10M,
    GROUP 3 '/private1/oracle/oradata/NLDAP/redo03.log' SIZE 10M
-- STANDBY LOGFILE
DATAFILE
    '/private1/oracle/oradata/NLDAP/system01.dbf',
    '/private1/oracle/oradata/NLDAP/sysaux01.dbf',
    '/private1/oracle/oradata/NLDAP/users01.dbf',
    '/private1/oracle/oradata/NLDAP/dcm.dbf',
    '/private1/oracle/oradata/NLDAP/portal.dbf',
    '/private1/oracle/oradata/NLDAP/ptldoc.dbf',
    '/private1/oracle/oradata/NLDAP/ptlidx.dbf',
    '/private1/oracle/oradata/NLDAP/ptllog.dbf',
    '/private1/oracle/oradata/NLDAP/oca.dbf',
    '/private1/oracle/oradata/NLDAP/discopltml.dbf',
    '/private1/oracle/oradata/NLDAP/discopltml.dbf',
    '/private1/oracle/oradata/NLDAP/oss_sys01.dbf',
    '/private1/oracle/oradata/NLDAP/wcrsys01.dbf',

```

```

'/private1/oracle/oradata/NLDAP/uddisys01.dbf',
'/private1/oracle/oradata/NLDAP/b2b_dt.dbf',
'/private1/oracle/oradata/NLDAP/b2b_rt.dbf',
'/private1/oracle/oradata/NLDAP/b2b_idx.dbf',
'/private1/oracle/oradata/NLDAP/b2b_lob.dbf',
'/private1/oracle/oradata/NLDAP/bam.dbf',
'/private1/oracle/oradata/NLDAP/orabpel.dbf',
'/private1/oracle/oradata/NLDAP/attrsl_oid.dbf',
'/private1/oracle/oradata/NLDAP/battrsl_oid.dbf',
'/private1/oracle/oradata/NLDAP/gcatsl_oid.dbf',
'/private1/oracle/oradata/NLDAP/gdefaultl_oid.dbf',
'/private1/oracle/oradata/NLDAP/svrmgl_oid.dbf',
'/private1/oracle/oradata/NLDAP/ias_meta01.dbf',
'/private1/oracle/oradata/NLDAP/undotbs.dbf'
CHARACTER SET AL32UTF8
;

```

9. Copy the initialization parameter file `init$ORACLE_SID.ora` of the sponsor directory's database to `init$ORACLE_SID_NEW_DIR_DB.ora`. The default location of the initialization parameter file is `$ORACLE_HOME/dbs` on UNIX and `%ORACLE_HOME%\database` on Windows. In our example, copy `/private/oracle/app/oracle/product/OraHome_1/dbs/initLDAP.ora` to `/private/oracle/app/oracle/product/OraHome_1/dbs/initNLDAP.ora` as shown here:

```

$cd $ORACLE_HOME/dbs
$cp initLDAP.ora initNLDAP.ora

```

If you are using the server parameter file `spfile$ORACLE_SID.ora` or `spfile.ora` instead of an initialization parameter file, create an initialization parameter file from the server parameter file as shown in the following example:

```

$sqlplus /nolog
SQL> connect / as sysdba
SQL> create pfile from spfile

```

The previous example assumes that `spfile$ORACLE_SID.ora` is located in the default location `$ORACLE_HOME/dbs`. In our example, the previous step will create an `initLDAP.ora` file at

`/private/oracle/app/oracle/product/OraHome_1 from spfileLDAP.ora. If the server parameter file is not located in the default location, you must include the complete path, as shown in the following example:`

```

$sqlplus /nolog
SQL> connect / as sysdba
SQL> create pfile='/private/oracle/initLDAP.ora' from
spfile=/private/oracle/initLDAP.ora

```

After you create the initialization file parameter file, create a copy of it as explained at the beginning of this step.

10. In the new initialization parameter file, make following changes:
 - a. Comment the parameter `JOB_QUEUE_PROCESSES`.
 - b. Change the parameter `dbname` from `LDAP` to `NLDAP`.
 - c. If the new site's domain name is different from the sponsor directory's domain name, alter the parameter `db_domain` also.
 - d. Alter the location of following parameters to point to location of the new site.

```
background_dump_dest
core_dump_dest
user_dump_dest
control_files
db_recovery_file_dest
```

- e. In addition to the parameters listed in step d, if your initialization parameter file has any parameters that are node specific, such as DB_RECOVERY_FILE_DEST and DB_CREATE_FILE_DEST, alter those parameters as well.

In our example, the initialization parameter file `initNLDAP.ora` will look like this after these modifications:

```
*.aq_tm_processes=1
*.background_dump_dest='/privatel/oracle/app/oracle/admin/NLDAP/bdump'
*.compatible='10.1.0.2.0'
*.control_files='/privatel/oracle/app/oracle/admin/NLDAP/control01.ctl',
                '/privatel/oracle/app/oracle/admin/NLDAP/control02.ctl',
                '/privatel/oracle/app/oracle/admin/NLDAP/control03.ctl'
*.core_dump_dest='/privatel/oracle/app/oracle/admin/NLDAP/cdump'
*.db_block_size=8192
*.db_cache_size=50331648
*.db_domain='acme.com'
*.db_file_multiblock_read_count=16
*.db_name='NLDAP'
*.db_recovery_file_dest='/private/oracle1/app/oracle/flash_recovery_area'
*.db_recovery_file_dest_size=2147483648
*.dispatchers='(PROTOCOL=TCP)(PRE=oracle.aurora.server.GiopServer)',
                '(PROTOCOL=TCP)(PRE=oracle.aurora.server.SGiopServer)'
*.java_pool_size=67108864
##.job_queue_processes=5
*.large_pool_size=8388608
*.max_commit_propagation_delay=0
*.open_cursors=300
*.pga_aggregate_target=33554432
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.sessions=400
*.shared_pool_size=150994944
*.undo_management='AUTO'
*.undo_tablespace='UNDOTBS'
*.user_dump_dest='/privatel/oracle/app/oracle/admin/NLDAP/udump'
```

11. Edit the `tnsnames.ora` file to include information connection details of to the new node. Refer to the following sample file:

```
LDAP.ACME.COM =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rst-sun)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ldap.acme.com)
    )
  )
)
NLDAP.ACME.COM =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = dsm-sun)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = nldap.acme.com)
    )
  )
)
```

```
)
```

- 12. 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:**

```
# 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/OraHome_1)
      (SID_NAME = NLDAP)
    )
    (SID_DESC =
      (SID_NAME = extproc)
      (ORACLE_HOME = /private1/oracle/app/oracle/product/OraHome_1)
      (PROGRAM = extproc)
    )
  )
)
STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = OFF
```

The default location of `tnsnames.ora` and `listener.ora` is the directory `$ORACLE_HOME/network/admin` on UNIX and the directory `ORACLE_HOME\network\admin` on Windows. It can also be in one of these locations:

- The directory specified by the `TNS_ADMIN` environment variable or registry value.
- On UNIX operating systems, the global configuration directory. For example, on Solaris, this directory is `/var/opt/oracle`.

- 13. Copy the updated `tnsnames.ora` file to the other nodes. You can copy them to other nodes either using FTP or another appropriate method.**

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`, `listener.ora`, and `sqlnet.ora` from the sponsor node to the new node.

- 14. Create an archive of all the data files and compress the archived file. For example:**

```
>> $ find / -name *.dbf -print \
      -exec tar rvf tar_file_name_with_absolute_path {} \;
```

This command will search for all files ending with extension `.dbf`, starting from the root directory. The assumption is that there is only one instance of the database server installed on the node and that data files end with the `.dbf` extension.

Compress the archive file:

```
>> $ compress tar_file_name_with_absolute_path
```

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 as to where you can 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. Stop application server control, DCM, opmn, database and listener services in the new node:

```
$> emctl stop iasconsole
$> $ORACLE_HOME/dcm/bin/dcmctl stop
$> $ORACLE_HOME/opmn/bin/opmnctl stopall
$> sqlplus "/ as sysdba"
SQL> shutdown immediate;
SQL> exit
$> lsnrctl [listener_name] stop
```

2. Copy the initialization parameter file `initNLDAP.ora` from the sponsor node (`rst-sun`) to the new node under the UNIX directory `$ORACLE_HOME/dbs` using FTP or another appropriate tool. Ensure that the contents of the copied file `initNLDAP.ora` are valid after copying.
3. Ensure that following files do not exist in the directory `$ORACLE_HOME/dbs` on UNIX or `ORACLE_HOME\dbs` in Windows:

- `spfileNLDAP.ora`
- `spfile.ora`

If either of these files exists, the Oracle database will use that file instead of the `initNLDAP.ora` file you copied from sponsor node.

4. Copy the archived file created in step 14 of steps to be performed at the sponsor node procedure, using FTP or another appropriate tool. The following example uses FTP tool to copy the archived file from `rst_sun`.

```
$ ftp
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.

Extract the archived file on the new node. For example:

```
$ unzip oradb.tar.Z
$ tar xvf oradb.tar
```

Ensure that the data files are extracted to correct directory. In our example, it is /private1/oracle/oradata/NLDAP

- Using FTP or another appropriate tool, copy the file `newdb.sql` you created in Step 6 of "[Tasks To Be Performed on the Sponsor Node](#)" on page I-4. For example:

```
$ cd /private1/oracle/app/oracle/admin/NLDAP/udump
$ ftp
ftp> open rst-sun
ftp> cd /private1/oracle/app/oracle/admin/LDAP/udump
ftp> mget newdb.sql
```

- At the UNIX shell prompt, set `ORACLE_BASE`, `ORACLE_HOME` and `ORACLE_SID` environment variables. For example (using the C shell):

```
$ setenv ORACLE_BASE /private1/oracle/app/oracle
$ setenv ORACLE_HOME /private1/oracle/app/oracle/product/OraHome_1
$ setenv ORACLE_SID NLDAP
```

- In the same UNIX shell, execute `newdb.sql` using `SQL*Plus` as shown in the following example:

```
$ sqlplus /nolog
SQL> connect /as sysdba
SQL> @newdb.sql
SQL> shutdown normal
SQL>exit
```

- Edit the initialization parameter file `initNLDAP.ora` in the `$ORACLE_HOME/dbs` directory to uncomment the parameter `job_queue_processes`. Its value must be equal to or greater than the number of nodes in the Directory Replication Group.

- Start up the database and listener as follows:

```
$ sqlplus /nolog
SQL> connect /as sysdba
SQL> startup mount
SQL> alter database open resetlogs
SQL> exit
$ lsnrctl start
```

- Log in to the sponsor node and start up the database and listener on the sponsor node. In our example, the sponsor node is `rst-sun`.

```
$ telnet rst-sun
$ sqlplus /nolog
SQL> connect /as sysdba
SQL> startup
SQL> exit
$ lsnrctl start # By default listener name is LISTENER
$ exit
```

- Change the global database name of the new node.

```
SQL> connect /as sysdba
SQL> alter database rename global_name to NLDAP;
SQL> exit
```

- Add a temporary file to the tablespace using the following command:

```
SQL> connect /as sysdba
```

```
SQL> ALTER TABLESPACE TEMP ADD TEMPFILE temp01.dbf size 2000k;
SQL> exit
```

13. Delete the wallet files oidpwdlldap1 and oidpwr* at new node and reset ODS password

```
$ cd $ORACLE_HOME/ldap/admin
$ rm oidpwdlldap1 oidpwr*
```

14. Reset the password and start Oracle Internet Directory processes.

```
$ oidpasswd connect=nldap.acme.com create_wallet=true current_password=ods

$ oidmon connect=nldap.acme.com start
$ oidctl connect=nldap.acme.com server=oidldapd instance=1 start
```

15. Reset the ReplicaID at the new node. After the database copy, the replicaid in the database of the new node will be same as that of the sponsor node. Therefore, the new node's replicaid must be replaced. The new value of replicaid must be of the form *hostname_sid* where *hostname* is the host name (without the domain name) of the new node where Oracle Internet Directory server's repository will be running and *sid* is the ORACLE_SID of the new node database. In our example, the replicaid is dsm-sun_nldap. Ensure that all letters of the replicaid are in lower case. To reset the value of replicaid:

a. Create a file, chgrid.ldif, with the following contents:

```
dn:
changetype: modify
replace: orclreplicaid
orclreplicaid: dsm-sun_nldap
```

b. Using the ldapmodify tool, change the replicaid:

```
$ $ORACLE_HOME/bin/ldapmodify -p port#_of_ldap_server -h hostname \
-f chgrid.ldif
```

16. Rename the replica subentry. The replica subentry's DN is of the format

```
orclreplicaid=replicaid,cn=replication configuration
```

Since the replica id of the new node is changed in the previous step, this entry DN must be renamed. To rename it, execute the following command:

```
$ $ORACLE_HOME/bin/ldapmoddn -h New_node_hostname -p port \
-D "cn=orcladmin" -w $1 -r \
-b "orclreplicaid=sponsor_replicaid,cn=replication configuration" \
-R "orclreplicaid=new_replicaid" -N "cn=replication configuration"
```

17. In addition to renaming the replica subentry, change the orclreplicauri, orclreplicasecondaryuri and orclreplicastate attributes of the replica subentry. The orclreplicauri and orclreplicasecondaryuri attributes must be modified to contain the URI of the new node's ldap server. The orclreplicastate attribute must be set to 6, which will be used by remtool to identify that this a database copy-based addnode. To change the values, proceed as follows.

a. Create an LDIF file, modsubentry.ldif, with the following contents:

```
dn: orclreplicaid=new_replicaid,
cn=replication configuration
changetype: modify
```

```

replace: orclreplicauri
#Use your host name and port number
#where ldap server is listening
orclreplicauri: ldap://dsum-sun:389/
-
replace:orclreplicasecondaryuri
#Use your fully qualified host name and
#the port number where ldap server is listening
orclreplicasecondaryuri:
ldap://dsum-sun.acme.com:389/
-
replace:orclreplicastate
orclreplicastate: 6

```

- b. Using the `ldapmodify` tool, apply the changes to the directory:

```
$ ldapmodify -p port#_of_ldap_server -h hostname -f chgrid.ldif
```

Note: Set the `orclreplicastate` attribute of the configuration entry `orclreplicaid=replicaid,cn=replication` to 6 at the node that will act as the Remote Master Site. The previous example assumes that the new node will be the Remote Master Site and sets the `orclreplicastate` attribute to 6 there. If you want to use the new node as the Master Definition Site and the sponsor node as the Remote Master Site, set the `orclreplicastate` attribute to 6 at the sponsor node.

18. Stop Oracle Internet Directory processes.

```
oidmon connect=connect_string stop
```

19. Reset the replication DN password of the new node:

```
remtool -presetpwd -bind new_node_host:new_node_port
```

20. Clean up changelog tables at the new node.

```

$ sqlplus /nolog
SQL> connect ods/<ods password>;
SQL> truncate table ods.ods_chg_log;
SQL> truncate table ods.ods_chg_stat;
SQL> truncate table ods.asr_chg_log;

```

21. To configure Oracle Database Advanced Replication, if you are adding new node to an existing DRG, at the shell prompt, execute the following command:

```
$ remtool -addnode
```

To configure Oracle Database Advanced Replication, if you are creating a new DRG consisting of sponsor node and new node, at the shell prompt, execute the following command:

```
$ remtool -asrsetup
```

See Also: The "remtool" command-line tool reference in *Oracle Identity Management User Reference*

22. Start up the LDAP replication server on all the nodes, including the new node and sponsor node.

Note: The sponsor node might contain changelogs for operations that occurred before you performed the database copy procedure. If so, after the replication server is started, these changelogs will be propagated to the new node. Propagation will fail, however, because directory data in the new database copy node is already consistent with data in the sponsor node. As a result, these changelogs will end up in the Human Intervention Queue at the database copy node.

If you have created a new DRG consisting of the sponsor node and the new node, you can avoid these errors by truncating the `ods_chg_log` table at the sponsor node before starting the LDAP server at the sponsor node.

If you have added the new node to an existing DRG, do not truncate the table at sponsor node. Instead, let the changes end up in the Human Intervention Queue at the new node. Then the administrator should clean up the queue as described in "[Resolving Conflicts Manually in a Multimaster Replication Group](#)" in [Chapter 25](#).

23. Update the `OIDport` and `OIDsslport` parameters in the `$ORACLE_HOME/config/ias.properties` file with the non-SSL port and SSL port where the directory server of the new node is currently listening:

```
[ComponentConfig]
...
[InstallData]
...
OIDhost=dsm-sun
OIDport=current_non_ssl_port_of_ldap_server
OIDsslport=current_ssl_port_of_ldap_server
...
FarmAdminSupported=FALSE
```

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 Oracle Database 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

Supplier	Consumer	Change Number
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.

Troubleshooting Oracle Internet Directory

This appendix explains typical problems that you could encounter while running or installing Oracle Internet Directory. It contains these sections:

- [Problems and Solutions](#)
- [Need More Help?](#)

Problems and Solutions

This section describes common Oracle Internet Directory error messages, problems and solutions. It contains the following topics:

- [Installation Errors](#)
- [TCP/IP Problems](#)
- [Directory Server Error Messages and Causes](#)
- [Troubleshooting Password Policies](#)
- [Troubleshooting Directory Performance](#)
- [Troubleshooting Starting, Stopping, and Restarting of the Directory Server](#)
- [Troubleshooting Oracle Internet Directory Replication](#)
- [Troubleshooting SSL Setup](#)
- [Troubleshooting Change Log Garbage Collection](#)
- [Troubleshooting Dynamic Password Verifiers](#)
- [Troubleshooting Oracle Internet Directory Password Wallets](#)
- [Troubleshooting bulkload](#)

Installation Errors

During installation and configuration of the Oracle Database, Oracle recommends that you select the character set UTF-8 to avoid possible problems with multibyte characters.

TCP/IP Problems

TCP/IP bugs in the operating system can interfere with Oracle Internet Directory service.

Do Not Use TCP-Based Monitoring of Oracle Internet Directory Server Availability on Microsoft Windows 2003 Server

If you use the F5 load balancer for monitoring Oracle Internet Directory server availability, configure the load balancer to use LDAP- or HTTP-based monitoring, as described in the *Oracle Application Server High Availability Guide* section "Configuring A Load Balancer For OracleAS Cluster (Identity Management)." Using TCP-based monitoring might cause the service to become unavailable, due to an operating system bug on Windows 2003 Server.

Do Not Install DaimondCS Port Explorer

Oracle Internet Directory will not work if DaimondCS Port Explorer is installed on the system.

Directory Server 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.

Oracle Database Server Error Due to Schema Modifications

You get error **ORA-1562**

Problem

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.

Solution

To solve this, increase the size of the rollback segments in the database server.

Constraint Violation Error Due to Editing a User or Group or Creating a Realm

You get the following error in `oidldap*.log`:

```
ORA-01483: invalid length for DATE or NUMBER bind variable.
```

You may also see the following error on your screen:

```
LDAP: error code 19 - Constraint Violation
```

These errors might only occur intermittently.

Problem

If you loaded the OracleAS Metadata Repository into an Oracle 10g Database that uses the AL32UTF8 character set, you may encounter some errors when you try to edit a user or Group, or Create Identity Management Realms in Oracle Internet Directory. Editing a user includes editing attributes for an existing user.

Solution

As a workaround, you can wait a bit and try editing the user again.

Standard Error Messages Returned from Oracle Directory Server

[Table J-1](#) lists standard error messages and their causes. Oracle Internet Directory also returns other messages listed and described in "[Additional Directory Server Error Messages](#)" on page J-4.

Table J-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.
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 of 1000.
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	The requested bind method is not supported by the server. For example, SASL clients requesting Kerberos authentication from Oracle Internet Directory receive this error in response.
09—LDAP_PARTIAL_RESULTS	Server returned a referral.
10—LDAP_REFERRAL	Server returned a referral.
12—LDAP_UNAVAILABLE_CRITICAL_EXTENSION	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.
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 <i>attribute</i> syntax is invalid. In a search, the <i>filter</i> syntax is invalid.
32—LDAP_NO_SUCH_OBJECT	The base specified for the operation does not exist.
34—LDAP_INVALID_DN_SYNTAX	Error in the DN syntax.
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.

Table J-1 (Cont.) Standard Error Messages

Error	Cause
53—LDAP_UNWILLING_TO_PERFORM	General error, or server is in read-only mode.
65—LDAP_OBJECT_CLASS_VIOLATION	A change to the entry violates the object class 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.
81—LDAP_SERVER_DOWN	Cannot contact the directory 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
89—LDAP_PARAM_ERROR	Bad parameter to an LDAP routine
90—LDAP_NO_MEMORY	Out of memory

Additional Directory Server Error Messages

Table J-2 lists additional directory server 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 J-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.

Table J-2 (Cont.) Additional Error Messages

Error	Cause
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 and 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)
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.

Table J-2 (Cont.) Additional Error Messages

Error	Cause
Mandatory attribute <parameter> is not defined in Admin Domain <parameter>.	MUST refers to attribute not defined. (schema modification)
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 ldapmodify. 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)
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)

Table J-2 (Cont.) Additional Error Messages

Error	Cause
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.

Troubleshooting Password Policies

This section describes error messages and problems related to password policies.

Password Policy Error Messages

Table J-3 contains the error messages 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 J-3 Password Policy Violation Error Messages

Error Number	Exception	Comment or Resolution
9000	GSL_PWDEXPIRED_EXCP	User's password has expired.
9001	GSL_ACCOUNTLOCKED_EXCP	User account is locked.
9002	GSL_EXPIREWARNING_EXCP	User password will expire in <code>pwdexpirewarning</code> seconds. Please change your password now.
9003	GSL_PWDMINLENGTH_EXCP	User password is not the required number of characters long.
9004	GSL_PWDNUMERIC_EXCP	User password does not contain required numeric characters.
9005	GSL_PWDNULL_EXCP	User password is a null password, which is disallowed.
9006	GSL_PWDINHISTORY_EXCP	User's new password is the same as the old one, which is disallowed.
9007	GSL_PWDILLEGALVALUE_EXCP	User password is the same as your <code>orclpwdillegalvalues</code> , which is disallowed.
9008	GSL_GRACELOGIN_EXCP	User password has expired. User has <code>pwdgraceloginlimit</code> grace logins left.
9050	GSL_ACCTDISABLED_EXCP	User account has been disabled.

See Also: ["Managing Password Policies"](#) on page 15-4

Possible Password Policy Problems

This section describes some of the potential problems with password policies and the corresponding solutions.

PASSWORD POLICY ERROR You get the error: `PASSWORD POLICY ERROR :9000: GSL_PWDEXPIRED_EXCP.`

Problem

Beginning with Release 9.0.4, the `pwdmaxage` attributes of the password policies are defaulted to time value of 60 days.

Beginning with Release 9.0.4, the default value for Password Expiry Time is set to 5184000—that is, 60 days. After 60 days from your installation date, the password for the Oracle directory integration and provisioning server (and any other assigned

passwords) automatically expire. If you have Directory Synchronization or Provisioning running, the ODISRV process will attempt to process the active profiles. Soon after password expiration, this repeated trying causes the connector to exceed the max grace logins exceeded, and the account to become locked. A view of the `odisrv.trc` file for each profile shows: [LDAP: error code 49 - Password Policy Error:9000: GSL_PWDEXPIRED_EXCP:Your Password has expired. Please contact the Administrator to change your password.] along with Java errors.

Solution

Do the following:

1. Use `oidpasswd` utility to unlock the `orcladmin` account. You will be prompted for the OID password, this is the ODS password which by default is the same as the `ias_admin` password:

```
$ oidpasswd connect=asdb unlock_su_acct=true
OID DB user password:
OID super user account unlocked successfully.
```

This unlocks only the super user account, `cn=orcladmin`. Do not confuse this account with the realm-specific `orcladmin` account `cn=orcladmin,cn=users,dc=xxxxx,dc=yyyyy`. They are two separate accounts.

After you reset it, the super user account still cannot login to OracleAS Single Sign-On by using the `orcladmin` account until you perform the next step.

2. Launch the Oracle Directory Manager (must be a release 10g client) and navigate to Password Policy Management. You will see two entries: `cn=PwdPolicyEntry` and the password policy for your realm—for example, `password_policy_entry,dc=acme,dc=com`.

Edit each of these, changing the `pwdmaxage` attribute to an appropriate value:

- 5184000 = 60 days (default)
- 7776000 = 90 days
- 10368000 = 120 days
- 15552000 = 180 days
- 31536000 = 1 year
- 0 = never expire (Oracle Internet Directory 10.1.2.0 and 9.0.4.2.0)
- 4294967295 = never expire (Oracle Internet Directory 9.0.4.1.0 and 9.0.4.0.0)

Note: It is very important to change this value in both places.

3. Launch the Oracle Directory Manager and navigate to the realm-specific `orcladmin` account. Find the `userpassword` attribute and reset the value to something new. You should then be able to launch any Oracle component that uses OracleAS Single Sign-On and login as `orcladmin`.
4. Rerun the `odisrvreg` utility to reset the randomly generated password for Directory Integration and Provisioning. Make sure `odisrv` is down. For example:

```
$ odisrvreg -D cn=orcladmin -w welcome1 -p 3060
Already Registered...Updating DIS password...
DIS registration successful.
```

§

5. Launch Oracle Directory Manager, expand Server Management, select Integration Servers and reset the **UserPassword** field under the **General** tab of each active connector.

Troubleshooting Directory Performance

This section gives some quick pointers for common performance-related problems.

Poor LDAP Search Performance

LDAP search performance is poor.

Problem

Various problems.

Solution

Make sure that:

- Schema associated with the ODS user is ANALYZED
- For searches involving multiple filter operands, make sure that the order in which they are given goes from the most specific to the least specific. For example, `&(uid=john.doe)(objectclass=person)` is better than `&(objectclass=person)(uid=john.doe)`.

Poor LDAP Add or Modify Performance

LDAP add or modify performance is poor.

Problem

Various problems

Solution

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

When estimating the statistics, you can use the OID Database Statistics Collection tool to analyze the various database ODS schema objects.

Both the tracing functionality described in "[Using Debug Logging](#)" on page 10-1 and the database tracing event 10046 can assist you in diagnosing performance issues.

See Also: The "oidstats.sql" command-line tool reference in *Oracle Identity Management User Reference* for instructions on using the OID Database Statistics Collection tool

"[Optimizing Searches](#)" on page 21-9 for instructions on optimizing searches

MetaLink note 243006.1 on Oracle MetaLink, <http://metalink.oracle.com>, for information on performance issues with group entries

Troubleshooting Starting, Stopping, and Restarting of the Directory Server

To troubleshoot starting and stopping the directory server, you must know the purpose of each tool involved, how all the tools work together, and the overall process for starting and stopping the server.

See Also: [Chapter 21, "Tuning Considerations for the Directory"](#)

About the Tools for Starting, Stopping, and Restarting the Directory Server Instance

The tool used to start and stop the directory server as an Oracle Application Server component is OPMN. For information on troubleshooting OPMN, see the "Troubleshooting" appendix in *Oracle Process Manager and Notification Server Administrator's Guide*.

There are two tools used to start, stop, and restart directory server instances: OID Control Utility (OIDCTL) and OID Monitor (OIDMON).

OIDCTL When OIDCTL is executed, it connects to the database as user ODS. Depending on the options used in the command, it either inserts or updates rows into a table named ODS.ODS_PROCESS. If the `START` option is used, then a row is inserted. If either the `STOP` or `RESTART` option is used, then a row is updated.

The ODS.ODS_PROCESS table includes the following information:

- `instance`—The unique number of the instance, any value between 0 and 1000
- `pid`—Process identifier, which will be updated by OIDMON when the process is started
- `state`—The type of operation requested

The possible values for `state` are:

- 0=stop
- 1=start
- 2=running
- 3=restart
- 4=shutdown
- 5=failedover

Note: When OPMN is used to stop the directory server, the value for `state` is initially 4, that is, shutdown. However, once OPMN starts the directory server again, the `state` value becomes 2, that is, running.

OIDMON To start, stop, or restart a directory server instance, OIDMON must be running. At specified intervals, this daemon checks the value of the `state` column in the ODS.ODS_PROCESS table.

If it finds a row with `state=0`, then it reads the `pid` and stops the process.

If it finds one with `state=1` or `state=4`, then it starts a new process and updates the `pid` column with a new process identifier.

If it finds one with `state=2`, then it reads the `pid` and verifies that the process with that `pid` is running. If it is not running, then OIDMON starts a new process and updates the `pid` column with a new process identifier.

If it finds a row with `state=3`, then OIDMON reads the `pid`, stops the process, starts a new one, and updates the `pid` accordingly.

If OIDMON cannot start the server for some reason, it retries. If it is not running on a node in an Oracle Application Server Cluster (Identity Management) configuration, and it is still unsuccessful after 10 retries, it deletes the row from the `ODS.ODS_PROCESS` table. If OIDMON is running on a node in a Oracle Application Server Cluster (Identity Management) configuration, it retries 100 times.

If it is still unsuccessful, it pushes the request to another node.

In short, OIDCTL inserts and updates state information in the rows in the `ODS.ODS_PROCESS` table. OIDMON then reads that information and performs the specified task.

About the Processes Involved in Starting, Stopping, and Restarting the Directory Server

Starting, stopping and restarting the directory server involves a number of processes. OIDMON is one process. On Unix, it is called `oidmon`. In a Microsoft Windows environment, it is called `oidmon.exe`.

To start an instance, OIDMON checks the unique number in the `instance` column mentioned in the previous section. It then starts another process, namely, the listener/dispatcher, which is different from the Oracle Net Services listener process. It stores the process identifier for that new process in the `pid` column.

The listener/dispatcher, in turn, starts a number of server processes as defined in the configuration set entry. Note that these server processes are controlled by the listener/dispatcher and not by OIDMON. If one of these processes fails, then it is automatically restarted by the listener/dispatcher.

Together, the listener/dispatcher and the server processes constitute a directory server instance. On UNIX, this directory server instance is called `oidldapd`. On Microsoft Windows, they are called `oidldapd.exe`.

In short, there are at least three processes: one for OIDMON and at least two for the directory server itself. When all processes are running, you should see something like the following on UNIX computers:

```
% ps -ef|grep oid
root 12387 12381 0 Mar 28 ? 0:05 oidldapd -i 1 -conf 0 key=811436710
root 12381 1 0 Mar 28 ? 0:10 oidmon start
root 13297 1 0 Mar 28 ? 0:14 oidldapd
```

Another way to obtain server information is by running `ldapcheck`. When you do this, you may see something like this:

```
Checking Oracle Internet Directory Processes ...
Process oidmon is Alive as PID 12381
Process oidldapd is Alive as PID 12387
Process oidldapd is Alive as PID 13297
Not Running ---- Process oidrepld
```

Problems Starting, Stopping, and Restarting the Directory Server

This section describes some problems you might have when starting, stopping, or restarting the directory server.

OIDCTL or OIDMON fails Either OIDCTL or OIDMON can fail for a number of reasons.

Problem

Incorrect syntax

Solution

Verify that you are using the correct syntax as described in "Oracle Internet Directory Server Administration Tools" in *Oracle Identity Management User Reference*. Note that the correct value of the connect option when using OIDCTL is the TNS alias—that is, the connect string—and not a host name or other value. See Oracle MetaLink note 155790.1, on Oracle MetaLink, <http://metalink.oracle.com>.

Problem

The Oracle Internet Directory-designated database is not running.

The Oracle Net Services configurations are incorrect.

Solution

Verify that the Oracle Internet Directory-designated database and the Oracle Net Services components are correctly configured and running. To do this, see if you can connect to the database by using SQL*Plus that is installed in the same *ORACLE_HOME* as OIDCTL. Log in as *ODS/ods_password@tns_alias* where *tns_alias* is the same as that used in the connect option with OIDCTL. See Oracle MetaLink note 155790.1, on Oracle MetaLink, <http://metalink.oracle.com>.

Problem

LDAP name resolution requires two instances of Oracle Internet Directory, but only one is running.

Solution

Verify that the value of the *DIRECTORY_SERVERS* parameter in the file *ldap.ora* is different from that specified in *NAMES.DIRECTORY_PATH* in the file *sqlnet.ora*. Both of these files are found in *ORACLE_HOME/network/admin*. If everything is working correctly, then selecting from *ODS.ODS_PROCESS* retrieves rows with state values described in "OIDCTL" on page J-10. See Oracle MetaLink note 155790.1, on Oracle MetaLink, <http://metalink.oracle.com>.

Information in *ODS.ODS_PROCESS* is correct, but processes still do not start.

When everything is working correctly, you should see at least three processes: one named *oidmon*, and at least two named *oidldapd*. *OIDMON* starts, stops, and restarts the server processes, and, because it does so at specified intervals, give it time to complete the requested operation.

Problem

Missing *oidldapd* file.

Solution

See *oidmon.log*. Look for the message: *No such file or directory*. To correct the problem, replace the executable file.

Problem

Wrong permissions on *oidldapd* executable file.

Solution

Look for the message `Exec of OIDLDAPD failed with error 13`. On UNIX, the `$ORACLE_HOME/bin/oidldapd` file must have the following permissions:

```
-rws--x--- 1 root dba 1691802 Jan 20 10:30 oidldapd
```

If the permissions are not correct, type the following, as root:

```
cd $ORACLE_HOME/bin
chown root:dba oidldapd
chmod 0710 oidldapd
chmod u+s oidldapd
```

Problem

You are running as a user with insufficient privilege

Solution

To confirm that this is the problem, see `oidmon.log`. Look for the message: `Permission denied or Open Wallet failed`. This happens if you are not running either as `root` or as the user who is in the `dba` group. To correct the problem, try again as the correct user.

Problem

A port is in use.

Solution

See `oidldapdXX.log`, where `XX` is the server instance number. Look for the message: `Bind failed on...` This indicates that the port that `oidldapd` is configured to listen on is in use by some other process. To determine which process is using the port, type:

```
netstat -a | grep portNum
```

If necessary, reconfigure the other process to use a different port or configure `oidldapd` to listen on another port by adding a `configset`. Remember that, by default, `oidldapd` listens on two ports, an SSL and non-SSL port.

Problem

On a cluster or Oracle Application Server Cluster (Identity Management) configuration, `OIDMON` pushes the server to another node in a cluster when it cannot start the server on the local node.

Solution

See `oidmon.log`. Look for the message: `gslsgfrPushServer: Could not start server on NodeA, trying to start on nodeNodeB`. To correct this problem, you must first determine why `OIDMON` cannot start the server on the local node.

Problem

A possible problem with Oracle Net Services or with the database itself.

Solution

See `oidmon.log`, `oidsrv.log`, `oidldapdxx.log`, where `xx` is the server instance number, and `oidrepdxx.log` where `XX` is Oracle directory integration and provisioning server instance number, for details about the problem.

A Row is Missing from ODS.ODS_PROCESS

Problem

In a cluster or Oracle Application Server Cluster (Identity Management) configuration, OIDMON successfully starts `oidldapd` on both nodes, but then initiates failover due to a time stamp difference.

Solution

See `oidmon.log`. On the node with the missing row, look for the message: Successfully failed over from *NodeA* to *NodeB*. On the other node, you will see an extra `oidldapd`. To correct the problem, adjust the system time on all nodes so that they are all within 250 seconds of one another.

Solution

See the trace files `oidldapdxx.log` where `xx` is the instance number, and `oidldapdxxsy.log` where `xx` is the instance number and `yy` is the process identifier. If the trace files do not give useful information or pointers to Oracle MetaLink documents, then do the following: (1) Stop the directory server processes; (2) Remove or rename old trace files; (3) Start OIDMON and a directory server with maximum debug level, namely, 11744051. Note that, to get the trace files, you must first stop, then start, the server; you cannot simply restart it. Investigate the new trace files, and, if needed, log an iTAR with Oracle Support Services and upload the trace files to the iTAR. See Oracle MetaLink note 155790.1, on Oracle MetaLink, <http://metalink.oracle.com>.

See Also: "How Failover Works in an Oracle Application Server Cluster (Identity Management) Environment" in *Oracle Application Server High Availability Guide* for more information on failover.

OIDCTL Error No processes are running, but using OIDCTL gives an error saying that the specified instance is already in use

Problem

This can occur, for example, after a machine restart when OIDMON is not running.

Solution

Start OIDMON, which, in turn, starts the directory server. See Oracle MetaLink note 155790.1, on Oracle MetaLink, <http://metalink.oracle.com>.

Solution

Use the `stop` option of OIDCTL to stop the specified instance. See Oracle MetaLink note 155790.1, on Oracle MetaLink, <http://metalink.oracle.com>.

Solution

If the directory server fails to start, you can override all user-specified configuration parameters to start it and then return the configuration sets to a workable state by using the `ldapmodify` operation. Use command-line options to `oidctl` to start the server with different configuration values, overriding any defined configuration sets

except for the values in `configset0`. Do not modify `configset0` because this technique relies on its minimal, default contents.

Solution

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

See Also: The "oidctl" command-line tool reference in *Oracle Identity Management User Reference* for more information on failover.

Troubleshooting Oracle Internet Directory Replication

This section discusses directory replication problems.

Whenever you investigate a replication problem, be sure to consult the log files `$ORACLE_HOME/ldap/oidrepld00.log` and `oidldapdxx.log` for information.

The replication server supports multiple debugging levels. To turn on replication debugging, specify the `-d decimal_debug_level` flag when you start the server. For example:

```
oidctl server=oidrepld connect=connect_string instance=instance_number \
      flags="-h host -p port -d decimal_debug_level"
```

Note: Turning on debugging will affect replication performance.

See Also: [Chapter 10, "Logging, Auditing, and Monitoring the Directory"](#) for more information about debugging.

Replication Server Does Not Start

There are several problems that can prevent the replication server from starting.

Problem

Invalid `oidctl` syntax

Solution

Use the following syntax to start the replication server.:

```
oidctl server=oidrepld connect=connect_string instance=instance_number \
      flags="-h host -p port"
```

Problem

Oracle Internet Directory is not running at the host and port you specified on the command line when you attempted to start the replication server. This caused the anonymous bind to the target Oracle Internet Directory to fail.

Solution

Make sure the target Oracle Internet Directory is up and running at the specified host and port.

Problem

The replication server is attempting to bind to the host and port specified in either the `orclreplicaprimeryurl` or the `orclreplicasecondaryurl` attribute of the Replica entry, but Oracle Internet Directory is running at a different host or port.

Solution

If you decide to run Oracle Internet Directory at a different host or port, add the new information to the `orclreplicasecondaryurl` attribute of the replica entry, as follows:

1. Prepare a modification file, `mod.ldif`. For example, to change to host `my.us.oracle.com` and port `4444`, you would specify:

```
dn: orclreplicaid=replica_ID, cn=replication configuration
changetype: modify
add: orclreplicasecondaryurl
orclreplicasecondaryurl: ldap://my.us.oracle.com:4444/
```

2. Run:

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

Problem

The `ReplBind` credential in the replication wallet `$ORACLE_HOME/ldap/admin/oidrORACLE_SID` is corrupt or invalid. That is, the password stored in the wallet is not the same as the password that is stored in the directory, or the wallet does not exist. This causes the replication bind to fail and the replication server to exit with an error.

You might see messages similar to this example in the file `oidrepldXX.log`:

```
2005/07/21:11:13:28 * gslrcfdReadReplDnPswd:Error reading repl passwd
2005/07/21:11:13:28 * gslrcfcReadReplConfig:Error found.
2005/07/21:11:13:28 * Failed to read replication configuration information.
```

Solution

Use `remtool` to fix the replication bind credential in the replication wallet or to synchronize between Oracle Internet Directory and the replication wallet.

- `remtool -pchgpwd` changes the password of the replication dn of a replica. Use this option if you know the current replication DN password stored in the directory and you want to change it both in the directory and in the wallet.
- `remtool -presetpwd` resets the password of the replication dn of a replica. Use this option if you know the current replication DN password stored in the directory and you want to change it both in the directory and in the wallet.
- `remtool -pchgwlpwd` changes password of replication dn of a replica only in the wallet. Use this option if you know the replication DN password stored in the directory but you are not sure whether the wallet has the correct password or you want to create the wallet file.

All of these options will create a wallet if one does not already exist.

See Also:

- The "remtool" command-line tool reference in *Oracle Identity Management User Reference* for more information about using remtool
- The "oidpasswd" command-line tool reference in *Oracle Identity Management User Reference* for more information about using oidpasswd

Repository Creation Assistant Error**Problem**

When you use the Oracle Application Server tool RepCA to load Oracle Internet Directory schema into an existing Oracle 10.1.0.3 Database, you might see the following error message in the \$ORACLE_HOME/assistants/repca/log/repca*log file:

```
SP2-0332: Cannot create spool file.
```

Solution

This error message can be ignored.

Errors in Replication Bootstrap

A number of errors can occur in replication bootstrap.

Problem

Some of the naming contexts failed to be bootstrapped.

Solution

Identify the naming contexts that failed to be bootstrapped, and use the `oidreconcile` tool to reconcile them. Then resume replication by setting the consumer's replica state to ONLINE mode

Problem

Various causes.

Solution

Identify the cause of the bootstrap failure and fix the cause, then restart bootstrapping by setting consumer's replica state to BOOTSTRAP mode.

Solution

To determine the exact cause of the error, examine the log file `oidldapdxx.log`. Look for error messages like those in the following example:

```
2004/09/14:12:57:23 * Starting OIDREPLD against dlsun1418:4444...
2004/09/14:12:57:25 * Starting scheduler...
2004/09/14:12:57:26 * Start to BootStrap from supplier=dlsun1418_replica to
consumer=dlsun1418_replica2
2004/09/14:12:57:27 * gslrbssSyncDIT:Replicating namingcontext=cn=oraclecontext
.....
2004/09/14:12:58:21 * gslrbssSyncDIT:Sync done successfully for namingctx:
cn=oraclecontext, 222 entries matched
2004/09/14:12:58:21 * gslrbssSyncDIT:Replicating namingcontext=cn=joe smith .....
```

```
2004/09/14:12:58:23 * BootStrap failure when adding DN=cn=Joe Smith,
server=dlsun1418_replica2,err=Constraint violation.
2004/09/14:12:58:23 * gslrbssSyncDIT:Sync failed for namingctx: cn=joe smith, only
1 entries retrieved
2004/09/14:12:58:23 * gslrbssSyncDIT:Replicating
namingcontext=cn=oracleschemaversion .....
2004/09/14:12:58:25 * gslrbssSyncDIT:Sync done successfully for namingctx:
cn=oracleschemaversion, 10 entries matched
2004/09/14:12:58:51 * gslrbsbBootStrap: Failure occurred when bootstrapping 1 out
of 3 namingcontext(s) from the supplier
```

Identify the cause of the bootstrap failure and fix it. You can identify the naming contexts that caused the problem, then use `oidreconcile` to compare and reconcile the naming contexts. Once you have resolved the problem, start bootstrapping again by starting the Oracle Internet Directory replication server.

Problem

The Oracle Internet Directory server was shut down during the bootstrapping

Solution

Make sure both the supplier Oracle Internet Directory and the consumer Oracle Internet Directory servers are up and running during replication bootstrapping.

Problem

Some of the entries being bootstrapped cannot be applied at the consumer due to a constraint violation.

Solution

Make sure the Oracle Internet Directory schema of the consumer are synchronized with those of the supplier before starting replication bootstrap. When you add an LDAP replica, `remtool` ensures that the Oracle Internet Directory schema on the consumer replica are synchronized with those on the supplier replica.

Problem

Improper replication filtering during bootstrapping. Replication supports excluding one or more attributes during bootstrapping. However, if a mandatory attribute of an entry is configured to be excluded, that entry cannot be applied at the consumer due to an objectclass violation.

Solution

Follow the replication naming context configuration rules in [Chapter 25, "Oracle Internet Directory Replication Administration"](#) to configure replication filtering properly.

If you are debugging LDAP replication, you should become familiar with the LDAP replica states. If LDAP-based replication is configured, when the replication server starts, it reads the replica state from the local replica. The replication server behaves differently, depending upon the local replica state. LDAP replication errors appear in `oidldapdxx.log`

See Also: [Appendix H, "LDAP Replica States"](#)

Problem

When you restart the replication server after the replication server failed to bootstrap a naming context having more than 5000 entries, you may see error messages similar to this in the log file `oidrepld00.log`:

```
2005/04/05:13:21:55 * gslrbssSyncDIT:Replicating namingcontext=dc=com .....
2005/04/05:15:36:09 * gslrbssSyncDIT:Subtree delete on dc=com failed.
Error=DSA is unwilling to perform
2005/04/05:15:36:09 * gslrbssSyncDIT:Sync failed for namingctx: dc=com, only
0 entries retrieved
```

The replication server performs two steps during bootstrap operation. First, in the consumer, it deletes the naming contexts that it has to bootstrap. Second, it copies entries belonging to those naming contexts from supplier to consumer. Deletion by the replication server of a naming context having several thousands of entries results in a big transaction. The undo tablespace needs to have sufficient space to accommodate a big transaction. If the database's undo tablespace does not have sufficient space, it will result in an ORA-30036 error.

Solution

Either have the database administrator add more space to the undo tablespace, or use the `bulkdelete` tool to delete the required naming context before you start the replication server.

Changes Are Not Replicated

Changes are not replicated from one node to another.

Problem

The replication server has run out of table space

Solution

Look for the following message in the server log:

```
OCI Error ORA-1653 : ORA-01653: unable to extend table ODS.ASR_CHG_LOG by 8192 in
tablespace OLTS_DEFAULT
```

Extend the table space and investigate why the table space keeps growing.

Problem

The target Oracle Internet Directory server is down.

Solution

Restart the target Oracle Internet Directory server.

Problem

Various causes

Solution

Make sure the replication server is started on all nodes, in multi-master replication, and at the consumer node in single-master or fan-out replication.

For multi-master Oracle Database Advanced Replication, use `remtool` to diagnostic and fix problems.

- `remtool -asrverify` verifies the correctness of a DRG setup and reports problems.
- `remtool -asrrectify` verifies the correctness of a DRG setup, reports problems, and attempts to rectify the problems.

Check the replication log and LDAP log for error messages and fix the cause of the error after investigation.

See Also: The "remtool" command-line tool reference in *Oracle Identity Management User Reference* for more information about using `remtool`.

Replication Stops Working

Problem

Data is not replicated between the replicas. In some cases, a working replication setup stops working after OID Human Intervention Queue entries are applied to one of the nodes. In other cases, adding or deleting a new replica causes problems or failures.

Problem

Various causes

Solution

See the following Oracle MetaLink notes on Oracle MetaLink, <http://metalink.oracle.com>:

Note 171693.1, "Resolving Conflicts"

Note 122039.1, "Troubleshooting Basics for Advanced Replication"

Note 213910.1, "Debugging OID Replication when ASR_CHG_LOG Never Gets Populated."

You can search for Oracle MetaLink notes by entering a term such as "replication" into the search box.

Troubleshooting SSL Setup

Describe symptom

Problem

Setting up Oracle Internet Directory for one-way LDAP connections over SSL fails.

Solution

Do not set up the SSL port of configset 0 with wallet mode 2 or 3. If you do, you will break Oracle Delegated Administration Services and other services and applications that expect to communicate with Oracle Internet Directory on the encrypted SSL port.

To correctly configure and test Oracle Internet Directory for SSL, follow the instructions in Oracle Metalink note 178714.1, on Oracle MetaLink, <http://metalink.oracle.com>. Also see the SSL section of the tutorial "Getting Started with Oracle Internet Directory" at http://www.oracle.com/technology/obe/obe_as_10g.

This section discusses possible problems when configuring SSL

Troubleshooting Change Log Garbage Collection

Both replication and Oracle Directory Integration and Provisioning use change logs to propagate information from a supplier directory to a consumer directory. All change logs are stored in the table `ods_chg_log`. In addition, replication change logs are stored in `asr_chg_log`.

This section discusses possible problems you might encounter with change log garbage collection.

Change Logs Are Not Purged

Change logs grow very large.

Problem

Change logs are not being purged due to a replication issue. For example, if a replication server has been down for a few days, replication change logs will not be purged because they are needed for replication recovery.

Solution

Resolve the replication issue. See ["Troubleshooting Oracle Internet Directory Replication"](#) on page J-15".

Problem

The attribute `orclpurgetargetage` is set too high and there are one or more enabled but inactive change log subscribers that do not update `orclLastAppliedChangeNumber` in their subscriber profiles. Change number-based purging won't purge change logs that are not yet consumed and time-based purging won't purge them because they're not old enough.

Solution

Set the attribute `orclpurgetargetage` to a smaller value so that change logs are purged sooner.

Solution

Disable inactive changelog subscribers so that change logs are purged by change log number-based purging. Locate such enabled but inactive subscriber profiles by examining the `orclLastAppliedChangeNumber` in all subscriber profiles by typing:

```
ldapsearch -v -p port -h host -D cn=orcladmin -w password \
  -b "cn=changelog subscriber,cn=oracle internet directory" \
  -s sub "objectclass=orclchangesubscriber" \
  orcllastappliedchangenumber orclsubscriberdisable
```

Look for an entry that has `orclSubscriberDisabled` equal to zero and an `orclLastAppliedChangeNumber` value that never changes. If such an entry exists, and the change log garbage collector's `orclpurgetargetage` is zero or greater, delete the value of `orclpurgetargetage`. When `orclpurgetargetage` is not defined or less than zero, the garbage collector will purge changes applied by the replication server, even if another subscriber has not updated its `orclLastAppliedChangeNumber`.

See Also: ■ ["Change Log Purging in Multimaster Replication"](#) on page 22-5.

Troubleshooting Dynamic Password Verifiers

[Table J-4](#) lists and describes the error messages for dynamic password verifiers.

Table J-4 Error Messages for Dynamic Password Verifiers

Error Code	Description
9022	A reversible encrypted password is missing from the user entry.
9023	The crypto type specified in the LDAP request control is not supported.
9024	The username parameter is missing from the LDAP request control.

If the directory is able to compare verifiers, and the comparison evaluates as false, the directory sends the standard error LDAP_COMPARE_FALSE to the client. Similarly, if the user being authenticated lacks a directory entry, the directory sends the standard error LDAP_NO_SUCH_OBJECT.

See Also: "Password Verifier Schema Elements" in *Oracle Identity Management User Reference*

Troubleshooting Oracle Internet Directory Password Wallets

The Oracle Internet Directory Server has two password wallets: `oidpwdlldap1` and `oidpwwdrSID`.

The `oidpwdlldap1` file contains the DN and password of an ODS user in encrypted format. The Oracle Internet Directory server uses the credential to connect to the backend database at startup time.

Oracle Internet Directory Server Does Not Start

Either `oidctl` or `opmn` fails to start an Oracle Internet Directory server instance.

Problem

The password stored in the `oidpwdlldap1` wallet is not synchronized with the ODS password in the backend database.

Solution

Try to connect to the database again using the `sqlplus` command:

```
sqlplus ods /ods_password@connect_string
```

If the connection succeeds, try to synchronize the password in the wallet with the ODS password by using the `oidpasswd` tool to create a new wallet with the correct password. For example:

```
>> oidpasswd connect=connect_string create_wallet=true
```

If the connection attempt fails, you must login into the backend database as a database administrator and change the ODS password by using the `sql` command:

```
>> alter user ods identified by some_new_password
```

Then try to create a new `oidpwdlldap1` to store the new password.

Solution

Try to start the Oracle Internet Directory server again.

The `oidpwrSID` file contains the DN and password of a replica DN in an encrypted format. The Oracle Internet Directory replication server uses the credential to connect to the Oracle Internet Directory server at startup time.

This is an example of a replication password wallet, `oidpwrSID`:

```
/-----BEGIN REPL CREDENTIAL:cn=replication dn,orclreplicaid=qdinh-sun_
adeldap,cn=replication configuration-----
ezNkZXMtY2JjLXBBrY3MlcGFkfQUnaz0Tsfzcp0nMlHcHAXchf5mJw+sb4y0bLvww3RvSg7H
S7/WskJB02fdSGRlmfWAV+6llkRQ26g==
-----END REPL CREDENTIAL:cn=replication dn,orclreplicaid=qdinh-sun_
adeldap,cn=replication configuration-----/
```

Password Not Synchronized

Either `oidctl` or `opmn` fails to start an Oracle Internet Directory server instance and the replication server log file `oidrepld00.log` reports that it is not able to bind.

Problem

The replica DN password stored in the `oidpwrSID` is not synchronized with the replica DN password in the Oracle Internet Directory server.

Solution

Try to connect to the Oracle Internet Directory server instance using the `ldapbind` command. Specify the replica DN stored in `oidpwrSID` and the replica DN password. For example:

```
>> ldapbind -h host -p port -D "cn=replication dn,orclreplicaid=qdinh-sun_adeldap,
cn=replication configuration" -w replica_dn_password
```

If the connection succeeds, then you can reset the password in the `oidpwrSID` wallet using `remtool` with the option `-pchgwlpwd`, which changes the password of the replication DN of a replica only in the wallet. If you do not remember the replication dn password, then you can reset it using `remtool` with the option `-prestpwd`, which resets the password of the replication dn of a replica.

After resetting the replication password wallet, restart the replication server instance again a using `opmnctl` or `oidctl`.

Troubleshooting bulkload**Problem**

The `bulkload` command-line tool might hang if you run it on Windows using a version of MKS Toolkit earlier than 8.6.

Solution

If a hang occurs during the `-check` or `-generate` phase of `bulkload`, you should cancel the `bulkload` command and repeat it.

If a hang occurs during the `-load` phase of `bulkload`, you should follow these steps:

1. Cancel the `bulkload` command.
2. Execute this command:

```
bulkload.sh -connect conn_str -recover
```

3. Repeat the original `bulkload` command.

Note: To update MKS Toolkit, visit <http://www.datafocus.com/>.

Need More Help?

You can find more solutions on Oracle MetaLink, <http://metalink.oracle.com>.
If you do not find a solution for your problem, log a service request.

See Also: *Oracle Application Server Release Notes*, available on the Oracle Technology Network:
<http://www.oracle.com/technology/documentation/index.html>

Glossary

3DES

See [Triple Data Encryption Standard \(3DES\)](#).

access control item (ACI)

Access control information represents the permissions that various entities or subjects have to perform operations on a given object in the directory. This information is stored in Oracle Internet Directory as user-modifiable operational [attributes](#), each of which is called an access control item (ACI). An ACI determines user access rights to directory data. It contains a set of rules for controlling access to entries (structural access items) and attributes (content access items). Access to both structural and content access items may be granted to one or more users or groups.

access control list (ACL)

A list of resources and the usernames of people who are permitted access to those resources within a computer system. In Oracle Internet Directory, an ACL is a list of [access control item \(ACI\) attribute values](#) that 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.

access control policy point (ACP)

A directory entry that contains access control policy information that applies downward to all entries at lower positions in the [directory information tree \(DIT\)](#). This information affects the entry itself and all entries below it. In Oracle Internet Directory, you can create ACPs to apply an access control policy throughout a [subtree](#) of your directory.

account lockout

A security feature that locks a user account if repeated failed logon attempts occur within a specified amount of time, based on security policy settings. Account lockout occurs in OracleAS Single Sign-On when a user submits an account and password combination from any number of workstations more times than is permitted by Oracle Internet Directory. The default lockout period is 24 hours.

ACI

See [access control item \(ACI\)](#).

ACL

See [access control list \(ACL\)](#).

ACP

See [access control policy point \(ACP\)](#).

administrative area

A [subtree](#) on a directory server whose entries are under the control of a single administrative authority. The designated administrator controls each [entry](#) in that administrative area, as well as the directory [schema](#), [access control list \(ACL\)](#), and [attributes](#) for those entries.

Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is a [symmetric cryptography](#) algorithm that is intended to replace [Data Encryption Standard \(DES\)](#). AES is a Federal Information Processing Standard (FIPS) for the encryption of commercial and government data.

advanced replication

See [Oracle Database Advanced Replication](#).

advanced symmetric replication (ASR)

See [Oracle Database Advanced Replication](#).

AES

See [Advanced Encryption Standard \(AES\)](#).

anonymous authentication

The process by which a 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 programming interface \(API\)](#).

application programming interface (API)

A series of software routines and development tools that comprise an interface between a computer application and lower-level services and functions (such as the operating system, device drivers, and other software applications). APIs serve as building blocks for programmers putting together software applications. For example, LDAP-enabled clients access Oracle Internet Directory information through programmatic calls available in the LDAP API.

application service provider

Application Service Providers (ASPs) are third-party entities that manage and distribute software-based services and solutions to customers across a wide area network from a central data center. In essence, ASPs are a way for companies to outsource some or almost all aspects of their information technology needs.

ASN.1

Abstract Syntax Notation One (ASN.1) is an International Telecommunication Union (ITU) notation used to define the syntax of information data. ASN.1 is used to describe structured information, typically information that is to be conveyed across some communications medium. It is widely used in the specification of Internet protocols.

ASR

See [Oracle Database Advanced Replication](#).

asymmetric algorithm

A **cryptographic algorithm** that uses different **keys** for **encryption** and **decryption**.

See also: **public key cryptography**.

asymmetric cryptography

See **public key cryptography**.

attribute

Directory attributes hold a specific data element such as a name, phone number, or job title. Each directory **entry** is comprised of 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 and Provisioning environment, a file that specifies attributes of interest in a connected directory.

attribute type

Attribute types specify information about a data element, such as the data type, maximum length, and whether it is single-valued or multivalued. The attribute type provides the real-world meaning for a value, and specifies the rules for creating and storing specific pieces of data, such as a name or an e-mail address.

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

Attribute values are the actual data contained within an **attribute** for a particular **entry**. For example, for the attribute type `email`, an attribute value might be `sally.jones@oracle.com`.

authentication

The process of verifying the identity claimed by an entity based on its credentials. Authentication of a user is generally based on something the user knows or has (for example, a password or a certificate).

Authentication of an electronic message involves the use of some kind of system (such as **public key cryptography**) to ensure that a file or message which claims to originate from a given individual or company actually does, and a check based on the contents of a message to ensure that it was not modified in transit.

authentication level

An OracleAS Single Sign-On parameter that enables you to specify a particular authentication behavior for an application. You can link this parameter with a specific **authentication plugin**.

authentication plugin

An implementation of a specific authentication method. OracleAS Single Sign-On has Java plugins for password authentication, digital certificates, Windows native authentication, and third-party access management.

authorization

The process of granting or denying access to a service or network resource. Most security systems are based on a two step process. The first stage is authentication, in which a user proves his or her identity. The second stage is authorization, in which a user is allowed to access various resources based on his or her identity and the defined [authorization policy](#).

authorization policy

Authorization policy describes how access to a protected resource is governed. Policy maps identities and objects to collections of rights according to some system model. For example, a particular authorization policy might state that users can access a sales report only if they belong to the sales group.

basic authentication

An [authentication](#) protocol supported by most browsers in which a Web server authenticates an entity with an encoded user name and password passed via data transmissions. Basic authentication is sometimes called plaintext authentication because the base-64 encoding can be decoded by anyone with a freely available decoding utility. Note that encoding is not the same as [encryption](#).

Basic Encoding Rules (BER)

Basic Encoding Rules (BER) are the standard rules for encoding data units set forth in [ASN.1](#). BER is sometimes incorrectly paired with ASN.1, which applies only to the abstract syntax description language, not the encoding technique.

BER

See [Basic Encoding Rules \(BER\)](#).

binding

In networking, binding is the establishment of a logical connection between communicating entities.

In the case of Oracle Internet Directory, binding refers to the process of authenticating to the directory.

The formal set of rules for carrying a [SOAP](#) message within or on top of another protocol (underlying protocol) for the purpose of exchange is also called a binding.

block cipher

Block ciphers are a type of [symmetric algorithm](#). A block cipher encrypts a message by breaking it down into fixed-size blocks (often 64 bits) and encrypting each block with a key. Some well known block ciphers include [Blowfish](#), [DES](#), and [AES](#).

See also: [stream cipher](#).

Blowfish

Blowfish is a [symmetric cryptography](#) algorithm developed by Bruce Schneier in 1993 as a faster replacement for [DES](#). It is a [block cipher](#) using 64-bit blocks and keys of up to 448 bits.

CA

See [Certificate Authority \(CA\)](#).

CA certificate

A **Certificate Authority (CA)** signs all certificates that it issues with its **private key**. The corresponding Certificate Authority's **public key** is itself contained within a certificate, called a CA Certificate (also referred to as a root certificate). A browser must contain the CA Certificate in its list of trusted root certificates in order to trust messages signed by the CA's private key.

cache

Generally refers to an amount of quickly accessible memory in your computer. However, on the Web it more commonly refers to where the browser stores downloaded files and graphics on the user's computer.

CBC

See **cipher block chaining (CBC)**.

central directory

In an Oracle Directory Integration and Provisioning environment, the directory that acts as the central repository. In an Oracle Directory Integration and Provisioning environment, Oracle Internet Directory is the central directory.

certificate

A certificate is a specially formatted data structure that associates a **public key** with the identity of its owner. A certificate is issued by a **Certificate Authority (CA)**. It contains the name, serial number, expiration dates, and public key of a particular entity. The certificate is digitally signed by the issuing CA so that a recipient can verify that the certificate is real. Most digital certificates conform to the **X.509** standard.

Certificate Authority (CA)

A Certificate Authority (CA) is a trusted third party that issues, renews, and revokes digital **certificates**. The CA essentially vouches for an entity's identity, and may delegate the verification of an applicant to a **Registration Authority (RA)**. Some well known Certificate Authorities (CAs) include Digital Signature Trust, Thawte, and VeriSign.

certificate chain

An ordered list of certificates containing one or more pairs of a user **certificate** and its associated **CA certificate**.

certificate management protocol (CMP)

Certificate Management Protocol (CMP) handles all relevant aspects of certificate creation and management. CMP supports interactions between **public key infrastructure (PKI)** components, such as the **Certificate Authority (CA)**, **Registration Authority (RA)**, and the user or application that is issued a certificate.

certificate request message format (CRMF)

Certificate Request Message Format (CRMF) is a format used for messages related to the life-cycle management of **X.509** certificates, as described in the **RFC 2511** specification.

certificate revocation list (CRL)

A Certificate Revocation List (CRL) is a list of digital **certificates** which have been revoked by the **Certificate Authority (CA)** that issued them.

change logs

A database that records changes made to a directory server.

cipher

See [cryptographic algorithm](#).

cipher block chaining (CBC)

Cipher block chaining (CBC) is a mode of operation for a [block cipher](#). CBC uses what is known as an initialization vector (IV) of a certain length. One of its key characteristics is that it uses a chaining mechanism that causes the decryption of a block of ciphertext to depend on all the preceding ciphertext blocks. As a result, the entire validity of all preceding blocks is contained in the immediately previous ciphertext block.

cipher suite

In [Secure Sockets Layer \(SSL\)](#), 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.

ciphertext

Ciphertext is the result of applying a [cryptographic algorithm](#) to readable data (plaintext) in order to render the data unreadable by all entities except those in possession of the appropriate [key](#).

circle of trust

A circle of trust is a [federation](#) of [service providers](#) and [identity providers](#) that have business relationships based on [Liberty Alliance](#) architecture and operational agreements, and with whom users can transact business in a secure and apparently seamless environment.

claim

A claim is a declaration made by an entity (for example, a name, identity, key, group, and so on).

client SSL certificates

A type of [certificate](#) used to identify a client machine to a server through [Secure Sockets Layer \(SSL\)](#) (client authentication).

cluster

A collection of interconnected usable whole computers that is used as a single computing resource. Hardware clusters provide high availability and scalability.

CMP

See [certificate management protocol \(CMP\)](#).

CMS

See [Cryptographic Message Syntax \(CMS\)](#).

code signing certificates

A type of [certificate](#) used to identify the entity who signed a Java program, Java Script, or other signed file.

cold backup

In Oracle Internet Directory, this refers to the procedure of adding a new **directory system agent (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 Oracle Internet 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.

confidentiality

In cryptography, confidentiality (also known as privacy) is the ability to prevent unauthorized entities from reading data. This is typically achieved through **encryption**.

configset

See **configuration set entry**.

configuration set entry

An Oracle Internet 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 **directory-specific entry (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 the Oracle 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 and Provisioning 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 **distinguished name (DN)** of the root of a **naming context**.

CRL

See **certificate revocation list (CRL)**.

CRMF

See **certificate request message format (CRMF)**.

cryptographic algorithm

A cryptographic algorithm is a defined sequence of processes to convert readable data (plaintext) to unreadable data (ciphertext) and vice versa. These conversions require some secret knowledge, normally contained in a **key**. Examples of cryptographic algorithms include **DES**, **AES**, **Blowfish**, and **RSA**.

Cryptographic Message Syntax (CMS)

Cryptographic Message Syntax (CMS) is a syntax defined in **RFC 3369** for signing, digesting, authenticating, and encrypting digital messages.

cryptography

The process of protecting information by transforming it into an unreadable format. The information is encrypted using a **key**, which makes the data unreadable, and is then decrypted later when the information needs to be used again. See also **public key cryptography** and **symmetric cryptography**.

dads.conf

A configuration file for Oracle HTTP Server that is used to configure a **database access descriptor (DAD)**.

DAS

See **Oracle Delegated Administration Services**. (DAS).

Data Encryption Standard (DES)

Data Encryption Standard (DES) is a widely used **symmetric cryptography** algorithm developed in 1974 by IBM. It applies a 56-bit key to each 64-bit block of data. DES and 3DES are typically used as encryption algorithms by **S/MIME**.

data integrity

The guarantee that the contents of the message received were not altered from the contents of the original message sent.

See also: **integrity**.

database access descriptor (DAD)

Database connection information for a particular Oracle Application Server component, such as the OracleAS Single Sign-On schema.

decryption

The process of converting the contents of an encrypted message (ciphertext) back into its original readable format (plaintext).

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 [directory information tree \(DIT\)](#).

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 realm location

An attribute in the [root Oracle Context](#) that identifies the root of the [default identity management realm](#).

Delegated Administration Services

See [Oracle Delegated Administration Services](#).

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.

DER

See [Distinguished Encoding Rules \(DER\)](#).

DES

See [Data Encryption Standard \(DES\)](#).

DIB

See [directory information base \(DIB\)](#).

Diffie-Hellman

Diffie-Hellman (DH) is a public key cryptography protocol that allows two parties to establish a shared secret over an unsecure communications channel. First published in 1976, it was the first workable public key cryptographic system.

See also: [symmetric algorithm](#).

digest

See [message digest](#).

digital certificate

See [certificate](#).

digital signature

A digital signature is the result of a two-step process applied to a given block of data. First, a [hash function](#) is applied to the data to obtain a result. Second, that result is

encrypted using the signer's **private key**. Digital signatures can be used to ensure integrity, message authentication, and non-repudiation of data. Examples of digital signature algorithms include **DSA**, **RSA**, and **ECDSA**.

Digital Signature Algorithm (DSA)

The Digital Signature Algorithm (DSA) is an **asymmetric algorithm** that is used as part of the Digital Signature Standard (DSS). It cannot be used for encryption, only for digital signatures. The algorithm produces a pair of large numbers that enable the authentication of the signatory, and consequently, the integrity of the data attached. DSA is used both in generating and verifying digital signatures.

See also: **Elliptic Curve Digital Signature Algorithm (ECDSA)**.

directory

See **Oracle Internet Directory**, **Lightweight Directory Access Protocol (LDAP)**, and **X.500**.

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 **DN**s of the entries.

directory integration and provisioning server

In an Oracle Directory Integration and Provisioning environment, the server that drives the synchronization of data between Oracle Internet Directory and a **connected directory**.

directory integration profile

In an Oracle Directory Integration and Provisioning environment, an entry in Oracle Internet Directory that describes how Oracle Directory Integration and Provisioning communicates with external systems and what is communicated.

Directory Manager

See **Oracle Directory Manager**.

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 Oracle Directory Integration and Provisioning 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 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.

directory-specific entry (DSE)

An entry specific to a directory server. Different directory servers may hold the same [directory information tree \(DIT\)](#) 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 user agent (DUA)

The software that accesses a directory service on behalf of the directory user. The directory user may be a person or another software element.

DIS

See [directory integration and provisioning server](#).

Distinguished Encoding Rules (DER)

Distinguished Encoding Rules (DER) are a set of rules for encoding [ASN.1](#) objects in byte-sequences. DER is a special case of [Basic Encoding Rules \(BER\)](#).

distinguished name (DN)

A [X.500](#) distinguished name (DN) is a unique name for a node in a directory tree. A DN is used to provide a unique name for a person or any other directory entry. A DN is a concatenation of selected [attributes](#) from each node in the tree along the path from the root node to the named entry's node. For example, in LDAP notation, the DN for a person named John Smith working at Oracle's US office would be: "cn=John Smith, ou=People, o=Oracle, c=us".

DIT

See [directory information tree \(DIT\)](#).

DN

See [distinguished name \(DN\)](#).

Document Type Definition (DTD)

A Document Type Definition (DTD) is a document that specifies constraints on the tags and tag sequences that are valid for a given [XML](#) document. DTDs follow the rules of Simple Generalized Markup Language (SGML), the parent language of XML.

domain component attribute

The domain component (dc) attribute can be used in constructing a [distinguished name \(DN\)](#) from a domain name. For example, using a domain name such as "oracle.com", one could construct a DN beginning with "dc=oracle, dc=com", and then use this DN as the root of its subtree of directory information.

DRG

See [directory replication group \(DRG\)](#).

DSA

See [Digital Signature Algorithm \(DSA\)](#) or [directory system agent \(DSA\)](#).

DSE

See [directory-specific entry \(DSE\)](#).

DTD

See [Document Type Definition \(DTD\)](#).

ECC

See [Elliptic Curve Cryptography \(ECC\)](#).

ECDSA

See [Elliptic Curve Digital Signature Algorithm \(ECDSA\)](#).

EJB

See [Enterprise Java Bean \(EJB\)](#).

Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) is an alternative to the [RSA](#) encryption system which is based on the difficulty of solving elliptic curve discrete logarithm problems rather than on factoring large numbers. Developed and marketed by Certicom, ECC is especially suitable for environments, such as wireless devices and PC cards, where computational power is limited and high speed is required. For any given key size (measured in bits) ECC provides more security (is harder to decrypt without the key) than RSA.

Elliptic Curve Digital Signature Algorithm (ECDSA)

The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analog of the [Digital Signature Algorithm \(DSA\)](#) standard. The advantages of ECDSA compared to RSA-like schemes are shorter key lengths and faster signing and decryption. For example, a 160 (210) bit ECC key is expected to give the same security as a 1024 (2048) bit RSA key, and the advantage increases as level of security is raised.

encryption

Encryption is the process of converting plaintext to ciphertext by applying a [cryptographic algorithm](#).

encryption certificate

An encryption certificate is a [certificate](#) containing a [public key](#) that is used to encrypt electronic messages, files, documents, or data transmission, or to establish or exchange a session key for these same purposes.

end-to-end security

This is a property of message-level security that is established when a message traverses multiple applications within and between business entities and is secure over its full route through and between the business entities.

Enterprise Java Bean (EJB)

Enterprise JavaBeans (EJBs) are a Java API developed by Sun Microsystems that defines a component architecture for multi-tier client/server systems. Because EJB systems are written in Java, they are platform independent. Being object oriented, they

can be implemented into existing systems with little or no recompiling and configuring.

Enterprise Manager

See [Oracle Enterprise Manager](#).

entry

An entry is a unique record in a directory that describes an object, such as a person. An entry consists of **attributes** and their associated **attribute values**, as dictated by the **object class** that describes that entry object. All entries in an LDAP directory structure are uniquely identified through their **distinguished name (DN)**.

export agent

In an Oracle Directory Integration and Provisioning environment, an agent that exports data out of Oracle Internet Directory.

export data file

In an Oracle Directory Integration and Provisioning 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 and Provisioning server. 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 Oracle Directory Integration and Provisioning.

external application

Applications that do not delegate authentication to the OracleAS Single Sign-On server. Instead, they display HTML login forms that ask for application user names and passwords. At the first login, users can choose to have the OracleAS Single Sign-On server retrieve these credentials for them. Thereafter, they are logged in to these applications transparently.

failover

The process of failure recognition and recovery. In an Oracle Application Server Cold Failover Cluster (Identity Management), 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.

Federal Information Processing Standards (FIPS)

Federal Information Processing Standards (FIPS) are standards for information processing issued by the US government Department of Commerce's National Institute of Standards and Technology (NIST).

federated identity management (FIM)

The agreements, standards, and technologies that make identity and entitlements portable across autonomous domains. FIM makes it possible for an authenticated user to be recognized and take part in personalized services across multiple domains. It avoids pitfalls of centralized storage of personal information, while allowing users to link identity information between different accounts. Federated identity requires two key components: trust and standards. The trust model of federated identity management is based on [circle of trust](#). The standards are defined by the [Liberty Alliance](#) Project.

federation

A federation is a group of entities (companies and organizations) that have a shared user base, and have agreed to provide identity and authorization tokens so that their users only have to logon once to access all of the services in their [circle of trust](#). Within the federation, at least one entity serves as the [identity provider](#) who is responsible for authenticating users. Entities that provide services to the user are referred to as [service providers](#).

filter

A filter is an expression that defines the entries to be returned from a request or search on a directory. Filters are typically expressed as DNs, for example: `cn=susie smith,o=acme,c=us`.

FIM

See [federated identity management \(FIM\)](#).

FIPS

See [Federal Information Processing Standards \(FIPS\)](#).

forced authentication

The act of forcing a user to reauthenticate if he or she has been idle for a preconfigured amount of time. Oracle Application Server Single Sign-On enables you to specify a global user inactivity timeout. This feature is intended for installations that have sensitive applications.

GET

An authentication method whereby login credentials are submitted as part of the login URL.

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.

global user inactivity timeout

An optional feature of Oracle Application Server Single Sign-On that forces users to reauthenticate if they have been idle for a preconfigured amount of time. The global user inactivity timeout is much shorter than the single sign-out session timeout.

globalization support

Multilanguage support for graphical user interfaces. Oracle Application Server Single Sign-On supports 29 languages.

globally unique user ID

A numeric string that uniquely identifies a user. A person may change or add user names, passwords, and distinguished names, but her globally unique user ID always remains the same.

grace login

A login occurring within the specified period before password expiration.

group search base

In the Oracle Internet Directory default **directory information tree (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.

See also: **hash function**.

hash function

In cryptography, a hash function or one-way hash function is an algorithm that produces a given value when applied to a given block of data. The result of a hash function can be used to ensure the integrity of a given block of data. For a hash function to be considered secure, it must be very difficult, given a known data block and a known result, to produce another data block that produces the same result.

Hashed Message Authentication Code (HMAC)

Hashed Message Authentication Code (HMAC) is a hash function technique used to create a secret hash function output. This strengthens existing hash functions such as MD5 and SHA. It is used in transport layer security (TLS).

HMAC

See **Hashed Message Authentication Code (HMAC)**.

HTTP

The Hyper Text Transfer Protocol (HTTP) is the protocol used between a Web browser and a server to request a document and transfer its contents. The specification is maintained and developed by the World Wide Web Consortium.

HTTP Server

See [Oracle HTTP Server](#).

httpd.conf

The file used to configure [Oracle HTTP Server](#).

iASAdmins

The administrative group responsible for user and group management functions in Oracle Application Server. The OracleAS Single Sign-On administrator is a member of the group iASAdmins.

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 infrastructure database

The database that contains data for OracleAS Single Sign-On and Oracle Internet Directory.

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.

identity provider

These are organizations recognized by the members of a [circle of trust](#) as the entity responsible for authenticating users and providing the digital identity information of

users to other parties in a **federation**. Identity providers enter into partnerships with service providers and provide services that follow agreed-upon practices set by all parties in a federation.

import agent

In an Oracle Directory Integration and Provisioning environment, an agent that imports data into Oracle Internet Directory.

import data file

In an Oracle Directory Integration and Provisioning environment, the file containing the data imported by an **import agent**.

infrastructure tier

The Oracle Application Server components responsible for identity management. These components are OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle Internet Directory.

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

In cryptography, integrity is the ability to detect if data has been modified by entities that are not authorized to modify it.

Internet Directory

See **Oracle Internet Directory**.

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.

J2EE

See **Java 2 Platform, Enterprise Edition (J2EE)**.

Java 2 Platform, Enterprise Edition (J2EE)

Java 2 Platform, Enterprise Edition (J2EE) is an environment for developing and deploying enterprise applications, defined by Sun Microsystems Inc. The J2EE platform consists of a set of services, application programming interfaces (APIs), and protocols that provide the functionality for developing multitiered, Web-based applications.

Java Server Page (JSP)

JavaServer Pages (JSP), a server-side technology, are an extension to the Java servlet technology that was developed by Sun Microsystems. JSPs have dynamic scripting capability that works in tandem with HTML code, separating the page logic from the static elements (the design and display of the page). Embedded in the HTML page, the Java source code and its extensions help make the HTML more functional, being used in dynamic database queries, for example.

JSP

See [Java Server Page \(JSP\)](#).

key

A key is a data structure that contains some secret knowledge necessary to successfully encrypt or decrypt a given block of data. The larger the key, the harder it is to crack a block of encrypted data. For example, a 256-bit key is more secure than a 128-bit key.

key pair

A [public key](#) and its associated [private key](#).

See also: [public/private key pair](#).

knowledge reference

The access information (name and address) for a remote [directory system agent \(DSA\)](#) and the name of the [directory information tree \(DIT\)](#) 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 \(LDAP\)](#).

LDAP connection cache

To improve throughput, the OracleAS Single Sign-On server caches and then reuses connections to Oracle Internet Directory.

LDAP Data Interchange Format (LDIF)

A common, text-based format for exchanging directory data between systems. The set of standards for formatting an input file for any of the LDAP command-line utilities.

LDIF

See [LDAP Data Interchange Format \(LDIF\)](#).

legacy application

Older application that cannot be modified to delegate authentication to the OracleAS Single Sign-On server. Also known as an [external application](#).

Liberty Alliance

The Liberty Alliance Project is an alliance of more than 150 companies, non-profit, and government organizations from around the globe. The consortium is committed to developing an open standard for federated network identity that supports all current

and emerging network devices. The Liberty Alliance is the only global body working to define and drive open technology standards, privacy, and business guidelines for [federated identity management \(FIM\)](#).

Lightweight Directory Access Protocol (LDAP)

A set of protocols for accessing information in directories. LDAP supports TCP/IP, which is necessary for any type of Internet access. Its framework of design conventions supports industry-standard directory products, such as Oracle Internet Directory. Because it is a simpler version of the [X.500](#) standard, LDAP is sometimes called X.500 light.

load balancer

Hardware devices and software that balance connection requests between two or more servers, either due to heavy load or failover. BigIP, Alteon, or Local Director are all popular hardware devices. Oracle Application Server Web Cache is an example of load balancing software.

logical host

In an Oracle Application Server Cold Failover Cluster (Identity Management), 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.

MAC

See [message authentication code \(MAC\)](#).

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 and Provisioning 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 \(MDS\)](#) 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.

MD2

Message Digest Two (MD2) is a message digest [hash function](#). The algorithm processes input text and creates a 128-bit [message digest](#) which is unique to the message and can be used to verify data integrity. MD2 was developed by Ron Rivest for RSA Security and is intended to be used in systems with limited memory, such as smart cards.

MD4

Message Digest Four (MD4) is similar to [MD2](#) but designed specifically for fast processing in software.

MD5

Message Digest Five (MD5) is a message digest [hash function](#). The algorithm processes input text and creates a 128-bit [message digest](#) which is unique to the message and can be used to verify data integrity. MD5 was developed by Ron Rivest after potential weaknesses were reported in [MD4](#). MD5 is similar to MD4 but slower because more manipulation is made to the original data.

MDS

See [master definition site \(MDS\)](#).

message authentication

The process of verifying that a particular message came from a particular entity.

See also: [authentication](#).

message authentication code (MAC)

The Message Authentication Code (MAC) is a result of a two-step process applied to a given block of data. First, the result of a [hash function](#) is obtained. Second, that result is encrypted using a [secret key](#). The MAC can be used to authenticate the source of a given block of data.

message digest

The result of a [hash function](#).

See also: [hash](#).

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.

middle tier

That portion of a OracleAS Single Sign-On instance that consists of the Oracle HTTP Server and OC4J. The OracleAS Single Sign-On middle tier is situated between the identity management infrastructure database and the client.

mod_osso

A module on the Oracle HTTP Server that enables applications protected by OracleAS Single Sign-On to accept HTTP headers in lieu of a user name and password once the user has logged into the OracleAS Single Sign-On server. The values for these headers are stored in the [mod_osso cookie](#).

mod_osso cookie

User data stored on the HTTP server. The cookie is created when a user authenticates. When the same user requests another application, the Web server uses the information in the mod_osso cookie to log the user in to the application. This feature speeds server response time.

mod_proxy

A module on the Oracle HTTP Server that makes it possible to use [mod_osso](#) to enable single sign-on to legacy, or [external applications](#).

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 references](#) (also called referrals) to subordinate naming contexts. It can range in size from a single entry to the entire [directory information tree \(DIT\)](#).

native agent

In an Oracle Directory Integration and Provisioning 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, for example:

```
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

Net Services

See [Oracle Net Services](#).

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.

non-repudiation

In cryptography, the ability to prove that a given **digital signature** was produced with a given entity's **private key**, and that a message was sent untampered at a given point in time.

OASIS

Organization for the Advancement of Structured Information Standards. OASIS is a worldwide not-for-profit consortium that drives the development, convergence and adoption of e-business standards.

object class

In LDAP, object classes are used to group information. Typically an object class models a real-world object such as a person or a server. Each directory entry belongs to one or more object classes. The object class determines the attributes that make up an entry. One object class can be derived from another, thereby inheriting some of the characteristics of the other class.

OC4J

See [Oracle Containers for J2EE \(OC4J\)](#).

OCA

See [Oracle Certificate Authority](#).

OCI

See [Oracle Call Interface \(OCI\)](#).

OCSP

See [Online Certificate Status Protocol \(OCSP\)](#).

OEM

See [Oracle Enterprise Manager](#).

OID

See [Oracle Internet Directory](#).

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 Internet Directory Server processes. It also controls the replication server if one is installed, and Oracle Directory Integration and Provisioning Server.

Online Certificate Status Protocol (OCSP)

Online Certificate Status Protocol (OCSP) is one of two common schemes for checking the validity of digital certificates. The other, older method, which OCSP has superseded in some scenarios, is [certificate revocation list \(CRL\)](#). OCSP is specified in [RFC 2560](#).

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.

See also: [hash function](#).

Oracle Application Server Single Sign-On

OracleAS Single Sign-On consists of program logic that enables you to log in securely to applications such as expense reports, mail, and benefits. These applications take two forms: [partner applications](#) and [external applications](#). In both cases, you gain access to several applications by authenticating only once.

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 Certificate Authority

Oracle Application Server Certificate Authority is a [Certificate Authority \(CA\)](#) for use within your Oracle Application Server environment. OracleAS Certificate Authority uses Oracle Internet Directory as the storage repository for certificates. OracleAS Certificate Authority integration with OracleAS Single Sign-On and Oracle Internet Directory provides seamless certificate provisioning mechanisms for applications relying on them. A user provisioned in Oracle Internet Directory and authenticated in OracleAS Single Sign-On can choose to request a digital certificate from OracleAS Certificate Authority.

Oracle CMS

Oracle CMS implements the IETF [Cryptographic Message Syntax \(CMS\)](#) protocol. CMS defines data protection schemes that allow for secure message envelopes.

Oracle Containers for J2EE (OC4J)

A lightweight, scalable container for [Java 2 Platform, Enterprise Edition \(J2EE\)](#).

Oracle Context

See [identity management realm-specific Oracle Context](#) and [root Oracle Context](#).

Oracle Crypto

Oracle Crypto is a pure Java library that provides core cryptography algorithms.

Oracle Database Advanced Replication

A feature in the Oracle Database that enables database tables to be kept synchronized across two Oracle databases.

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 and Provisioning

A collection of interfaces and services for integrating multiple directories by using Oracle Internet Directory and several associated plug-ins and connectors. A feature of Oracle Internet Directory that enables an enterprise to use an external user repository to authenticate to Oracle products.

Oracle Directory Integration and Provisioning Server

In an Oracle Directory Integration and Provisioning 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 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 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 HTTP Server

Software that processes Web transactions that use the Hypertext Transfer Protocol (HTTP). Oracle uses HTTP software developed by the Apache Group.

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 the Oracle Database.

Oracle Liberty SDK

Oracle Liberty SDK implements the [Liberty Alliance](#) Project specifications enabling federated single sign-on between third-party Liberty-compliant applications.

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 PKI SDK

Oracle PKI SDK implements the security protocols that are necessary within [public key infrastructure \(PKI\)](#) implementations.

Oracle SAML

Oracle SAML provides a framework for the exchange of security credentials among disparate systems and applications in an XML-based format as outlined in the [OASIS](#) specification for the [Security Assertions Markup Language \(SAML\)](#).

Oracle Security Engine

Oracle Security Engine extends Oracle Crypto by offering X.509 based certificate management functions. Oracle Security Engine is a superset of Oracle Crypto.

Oracle S/MIME

Oracle S/MIME implements the [Secure/Multipurpose Internet Mail Extension \(S/MIME\)](#) specifications from the [Internet Engineering Task Force \(IETF\)](#) for secure e-mail.

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*.

Oracle Web Services Security

Oracle Web Services Security provides a framework for authentication and authorization using existing security technologies as outlined in the [OASIS](#) specification for Web Services Security.

Oracle XML Security

Oracle XML Security implements the W3C specifications for XML Encryption and XML Signature.

OracleAS Portal

An OracleAS Single Sign-On [partner application](#) that provides a mechanism for integrating files, images, applications, and Web sites. The External Applications portlet provides access to external applications.

other information repository

In an Oracle Directory Integration and Provisioning environment, in which Oracle Internet Directory serves as the [central directory](#), any information repository except Oracle Internet Directory.

OWM

See [Oracle Wallet Manager](#).

partition

A unique, non-overlapping directory naming context that is stored on one directory server.

partner application

An Oracle Application Server application or non-Oracle application that delegates the authentication function to the OracleAS Single Sign-On server. This type of application spares users from reauthenticating by accepting **mod_osso** headers.

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#1

The Public Key Cryptography Standards (PKCS) are specifications produced by RSA Laboratories. PKCS#1 provides recommendations for the implementation of public-key cryptography based on the RSA algorithm, covering the following aspects: cryptographic primitives; encryption schemes; signature schemes; ASN.1 syntax for representing keys and for identifying the schemes.

PKCS#5

The Public Key Cryptography Standards (PKCS) are specifications produced by RSA Laboratories. PKCS#5 provides recommendations for the implementation of password-based cryptography.

PKCS#7

The Public Key Cryptography Standards (PKCS) are specifications produced by RSA Laboratories. PKCS #7 describes general syntax for data that may have cryptography applied to it, such as digital signatures and digital envelopes.

PKCS#8

The Public Key Cryptography Standards (PKCS) are specifications produced by RSA Laboratories. PKCS #8 describes syntax for private key information, including a private key for some public key algorithms and a set of attributes. The standard also describes syntax for encrypted private keys.

PKCS#10

The Public Key Cryptography Standards (PKCS) are specifications produced by RSA Laboratories. PKCS #10 describes syntax for a request for certification of a public key, a name, and possibly a set of attributes.

PKCS#12

The Public Key Cryptography Standards (PKCS) are specifications produced by RSA Laboratories. PKCS #12 describes a transfer syntax for personal identity information, including private keys, certificates, miscellaneous secrets, and extensions. Systems (such as browsers or operating systems) that support this standard allow a user to import, export, and exercise a single set of personal identity information—typically in a format called a **wallet**.

PKI

See **public key infrastructure (PKI)**.

plaintext

Plaintext is readable data prior to a transformation to ciphertext using encryption, or readable data that is the result of a transformation from ciphertext using decryption.

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.

policy precedence

In Oracle Application Server Certificate Authority (OCA), policies are applied to incoming requests in the order that they are displayed on the main policy page. When the OCA policy processor module parses policies, those that appear toward the top of the policy list are applied to requests first. Those that appear toward the bottom of the list are applied last and take precedence over the others. Only enabled policies are applied to incoming requests.

policy.properties

A multipurpose configuration file for Oracle Application Server Single Sign-On that contains basic parameters required by the single sign-on server. Also used to configure advanced features of OracleAS Single Sign-On, such as multilevel authentication.

POSIX

Portable Operating System Interface for UNIX. A set of programming interface standards governing how to write application source code so that the applications are portable between operating systems. A series of standards being developed by the [Internet Engineering Task Force \(IETF\)](#).

POST

An authentication method whereby login credentials are submitted within the body of the login form.

predicates

In Oracle Application Server Certificate Authority (OCA), a policy predicate is a logical expression that can be applied to a policy to limit how it is applied to incoming certificate requests or revocations. For example, the following predicate expression specifies that the policy in which it appears can have a different effect for requests or revocations from clients with DNs that include "ou=sales,o=acme,c=us":

```
Type=="client" AND DN=="ou=sales,o=acme,c=us"
```

primary node

In an Oracle Application Server Cold Failover Cluster (Identity Management), the cluster node on which the application runs at any given time.

See also: [secondary node](#).

private key

A private key is the secret key in a [public/private key pair](#) used in [public key cryptography](#). An entity uses its private key to decrypt data that has been encrypted with its [public key](#). The entity can also use its private key to create [digital signatures](#). The security of data encrypted with the entity's public key as well as signatures created by the private key depends on the private key remaining secret.

private key cryptography

See [symmetric cryptography](#).

profile

See [directory integration profile](#).

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.

provisioning

The process of providing users with access to applications and other resources that may be available in an enterprise environment.

provisioning agent

An application or process that translates Oracle-specific provisioning events to external or third-party application-specific events.

provisioning integration profile

A special kind of [directory integration profile](#) that describes the nature of provisioning-related notifications that Oracle Directory Integration and Provisioning sends to the directory-enabled applications.

proxy server

A server between a client application, such as a Web browser, and a real server. It intercepts all requests to the real server to see if it can fulfil the requests itself. If not, it forwards the request to the real server. In OracleAS Single Sign-On, proxies are used for load balancing and as an extra layer of security.

See also: [load balancer](#).

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

A public key is the non-secret key in a [public/private key pair](#) used in [public key cryptography](#). A public key allows entities to encrypt data that can only then be decrypted with the public key's owner using the corresponding [private key](#). A public key can also be used to verify digital signatures created with the corresponding private key.

public key certificate

See [certificate](#).

public key cryptography

Public key cryptography (also known as asymmetric cryptography) uses two keys, one public and the other private. These keys are called a key pair. The private key must be kept secret, while the public key can be transmitted to any party. The private key and

the public key are mathematically related. A message that is signed by a private key can be verified by the corresponding public key. Similarly, a message encrypted by the public key can be decrypted by the private key. This method ensures privacy because only the owner of the private key can decrypt the message.

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 key infrastructure (PKI)

A public key infrastructure (PKI) is a system that manages the issuing, distribution, and authentication of **public keys** and **private keys**. A PKI typically comprises the following components:

- A **Certificate Authority (CA)** that is responsible for generating, issuing, publishing and revoking digital certificates.
- A **Registration Authority (RA)** that is responsible for verifying the information supplied in requests for certificates made to the CA.
- A directory service where a **certificate** or **certificate revocation list (CRL)** gets published by the CA and where they can be retrieved by relying third parties.
- Relying third parties that use the certificates issued by the CA and the **public keys** contained therein to verify **digital signatures** and encrypt data.

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.

RC2

Rivest Cipher Two (RC2) is a 64-bit **block cipher** developed by Ronald Rivest for RSA Security, and was designed as a replacement for **Data Encryption Standard (DES)**.

RC4

Rivest Cipher Four (RC4) is a **stream cipher** developed by Ronald Rivest for RSA Security. RC4 allows variable key lengths up to 1024 bits. RC4 is most commonly used to secure data communications by encrypting traffic between Web sites that use the **Secure Sockets Layer (SSL)** protocol.

RDN

See **relative distinguished name (RDN)**.

readable data

Data prior to a transformation to ciphertext via encryption or data that is the result of a transformation from ciphertext via decryption.

realm

See **identity management realm**.

realm search base

An attribute in the **root Oracle Context** that identifies the entry in the **directory information tree (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**.

Registration Authority (RA)

The Registration Authority (RA) is responsible for verifying and enrolling users before a certificate is issued by a **Certificate Authority (CA)**. The RA may assign each applicant a relative distinguished value or name for the new certificate applied. The RA does not sign or issue certificates.

registry entry

An entry containing runtime information associated with invocations of Oracle Internet 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.

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.

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 **Oracle Database Advanced Replication**.

replica

Each copy of a **naming context** that is contained within a single server.

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.

RFC

The Internet Request For Comments (or RFC) documents are the written definitions of the protocols and policies of the Internet. The Internet Engineering Task Force (IETF) facilitates the discussion, development, and establishment of new standards. A

standard is published using the RFC acronym and a reference number. For example, the official standard for e-mail is RFC 822.

root CA

In a hierarchical [public key infrastructure \(PKI\)](#), the root [Certificate Authority \(CA\)](#) is the CA whose [public key](#) serves as the most trusted datum for a security domain.

root directory specific entry (DSE)

An entry storing operational information about the directory. The information is stored in a number of attributes.

root DSE

See [root directory specific entry \(DSE\)](#).

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.

RSA

RSA is a [public key cryptography](#) algorithm named after its inventors (Rivest, Shamir, and Adelman). The RSA algorithm is the most commonly used encryption and authentication algorithm and is included as part of the Web browsers from Netscape and Microsoft, and many other products.

RSAES-OAEP

The RSA Encryption Scheme - Optimal Asymmetric Encryption Padding (RSAES-OAEP) is a public key encryption scheme combining the [RSA](#) algorithm with the OAEP method. Optimal Asymmetric Encryption Padding (OAEP) is a method for encoding messages developed by Mihir Bellare and Phil Rogaway.

S/MIME

See [Secure/Multipurpose Internet Mail Extension \(S/MIME\)](#).

SAML

See [Security Assertions Markup Language \(SAML\)](#).

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 an Oracle Application Server Cold Failover Cluster (Identity Management), the cluster node to which an application is moved during a failover.

See also: [primary node](#).

secret key

A secret key is the **key** used in a **symmetric algorithm**. Since a secret key is used for both encryption and decryption, it must be shared between parties that are transmitting ciphertext to one another but must be kept secret from all unauthorized entities.

secret key cryptography

See **symmetric cryptography**.

Secure Hash Algorithm (SHA)

Secure Hash Algorithm (SHA) is a **hash function** algorithm that produces a 160-bit **message digest** based upon the input. The algorithm is used in the Digital Signature Standard (DSS). With the introduction of the Advanced Encryption Standard (AES) which offers three key sizes: 128, 192 and 256 bits, there has been a need for a companion hash algorithm with a similar level of security. The newer SHA-256, SHA-284 and SHA-512 hash algorithms comply with these enhanced requirements.

Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) is a protocol designed by Netscape Communications to enable encrypted, authenticated communications across networks (such as the Internet). SSL uses the **public key encryption** system from RSA, which also includes the use of a digital certificate. SSL provides three elements of secure communications: **confidentiality**, **authentication**, and **integrity**.

SSL has evolved into **Transport Layer Security (TLS)**. TLS and SSL are not interoperable. However, a message sent with TLS can be handled by a client that handles SSL.

Secure/Multipurpose Internet Mail Extension (S/MIME)

Secure/Multipurpose Internet Mail Extension (S/MIME) is an Internet Engineering Task Force (IETF) standard for securing MIME data through the use of **digital signatures** and **encryption**.

Security Assertions Markup Language (SAML)

Security Assertions Markup Language (SAML) is an **XML**-based framework for exchanging security information over the Internet. SAML enables the exchange of **authentication** and **authorization** information between various security services systems that otherwise would not be able to interoperate. The SAML 1.0 specification was adopted by **OASIS** in 2002.

server certificate

A **certificate** that attests to the identity of an organization that uses a secure Web server to serve data. A server certificate must be associated with a **public/private key pair** issued by a mutually trusted **Certificate Authority (CA)**. Server certificates are required for secure communications between a browser and a Web server.

service provider

These are organizations recognized by the members of a **circle of trust** as the entities that provide Web-based services to users. Service providers enter into partnerships with other service providers and identity providers with the goal of providing their common users with secure single sign-on between all parties of the **federation**.

service time

The time between the initiation of a request and the completion of the response to the request.

session key

A **secret key** 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.

Signed Public Key And Challenge (SPKAC)

Signed Public Key And Challenge (SPKAC) is a proprietary protocol used by the Netscape Navigator browser to request certificates.

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.

single sign-off

The process by which you terminate an OracleAS Single Sign-On session and log out of all active partner applications simultaneously. You can do this by logging out of the application that you are working in.

single sign-on (SSO)

A process or system that enables a user to access multiple computer platforms or application systems after being authenticated only once.

single sign-on SDK

Legacy APIs to enable OracleAS Single Sign-On partner applications for single sign-on. The SDK consists of PL/SQL and Java APIs as well as sample code that demonstrates how these APIs are implemented. This SDK is now deprecated and [mod_osso](#) is used instead.

single sign-on server

Program logic that enables users to log in securely to single sign-on applications such as expense reports, mail, and benefits.

SLAPD

Standalone LDAP daemon. An LDAP directory server service that is responsible for most functions of a directory except replication.

slave

See [consumer](#).

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.

SOAP

Simple Object Access Protocol (SOAP) is an [XML](#)-based protocol that defines a framework for passing messages between systems over the Internet via HTTP. A SOAP message consists of three parts — an envelope that describes the message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses.

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.

SPKAC

See [Signed Public Key And Challenge \(SPKAC\)](#).

sponsor node

In replication, the node that is used to provide initial data to a new node.

SSL

See [Secure Sockets Layer \(SSL\)](#).

stream cipher

Stream ciphers are a type of [symmetric algorithm](#). A stream cipher encrypts in small units, often a bit or a byte at a time, and implements some form of feedback

mechanism so that the key is constantly changing. **RC4** is an example of a stream cipher.

See also: **block cipher**.

subACLSubentry

A specific type of **subentry** that contains **access control list (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 CA

In a hierarchical **public key infrastructure (PKI)**, the subordinate **Certificate Authority (CA)** is a CA whose certificate signature key is certified by another CA, and whose activities are constrained by that other CA.

subordinate reference

A **knowledge reference** pointing downward in the **directory information tree (DIT)** to a **naming context** that starts immediately below an entry

subschema DN

The list of **directory information tree (DIT)** areas having independent **schema** definitions.

subSchemaSubentry

A specific type of **subentry** containing **schema** information.

subtree

A section of a directory hierarchy, which is also called a **directory information tree (DIT)**. The subtree typically starts at a particular directory node and includes all subdirectories and objects below that node in the directory hierarchy.

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.

success URL

When using Oracle Application Server Single Sign-On, the URL to the routine responsible for establishing the session and session cookies for an application.

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 **directory system agent (DSA)** that holds a naming context higher in the **directory information tree (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.

symmetric algorithm

A symmetric algorithm is a cryptographic algorithm that uses the same key for encryption and decryption. There are essentially two types of symmetric (or secret key) algorithms — **stream ciphers** and **block ciphers**.

symmetric cryptography

Symmetric cryptography (or shared secret cryptography) systems use the same key to encipher and decipher data. The problem with symmetric cryptography is ensuring a secure method by which the sender and recipient can agree on the secret key. If a third party were to intercept the secret key in transit, they could then use it to decipher anything it was used to encipher. Symmetric cryptography is usually faster than asymmetric cryptography, and is often used when large quantities of data need to be exchanged. **DES**, **RC2**, and **RC4** are examples of symmetric cryptography algorithms.

symmetric key

See **secret key**.

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.

think time

The time the user is not engaged in actual use of the processor.

third-party access management system

Non-Oracle single sign-on system that can be modified to use OracleAS Single Sign-On to gain access to Oracle Application Server applications.

throughput

The number of requests processed by Oracle Internet Directory for each unit of time. This is typically represented as "operations per second."

Time Stamp Protocol (TSP)

Time Stamp Protocol (TSP), as specified in RFC 3161, defines the participating entities, the message formats, and the transport protocol involved in time stamping a digital message. In a TSP system, a trusted third-party Time Stamp Authority (TSA) issues time stamps for messages.

TLS

See [Transport Layer Security \(TLS\)](#).

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.

Triple Data Encryption Standard (3DES)

Triple Data Encryption Standard (3DES) is based on the [Data Encryption Standard \(DES\)](#) algorithm developed by IBM in 1974, and was adopted as a national standard in 1977. 3DES uses three 64-bit long keys (overall key length is 192 bits, although actual key length is 56 bits). Data is encrypted with the first key, decrypted with the second key, and finally encrypted again with the third key. This makes 3DES three times slower than standard DES but also three times more secure.

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, trusted certificates come from a [Certificate Authority \(CA\)](#) you trust to issue user certificates.

trustpoint

See [trusted certificate](#).

TSP

See [Time Stamp Protocol \(TSP\)](#).

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.

URI

Uniform Resource Identifier (URI). A way to identify any point of content on the Web, whether it be a page of text, a video or sound clip, a still or animated image, or a program. The most common form of URI is the Web page address, which is a particular form or subset of URI called a **URL**.

URL

Uniform Resource Locator (URL). The address of a file accessible on the Internet. The file can be a text file, HTML page, image file, a program, or any other file supported by HTTP. The URL contains the name of the protocol required to access the resource, a domain name that identifies a specific computer on the Internet, and a hierarchical description of the file location on the computer.

URLC token

The OracleAS Single Sign-On code that passes authenticated user information to the **partner application**. The partner application uses this information to construct the session cookie.

user name mapping module

A OracleAS Single Sign-On Java module that maps a user **certificate** to the user's nickname. The nickname is then passed to an authentication module, which uses this nickname to retrieve the user's certificate from the directory.

user search base

In the Oracle Internet Directory default **directory information tree (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).

UTF-16

16-bit encoding of **Unicode**. The Latin-1 characters are the first 256 code points in this standard.

verification

Verification is the process of ensuring that a given **digital signature** is valid, given the **public key** that corresponds to the **private key** purported to create the signature and the data block to which the signature purportedly applies.

virtual host

A single physical Web server machine that is hosting one or more Web sites or domains, or a server that is acting as a proxy to other machines (accepts incoming requests and reroutes them to the appropriate server).

In the case of OracleAS Single Sign-On, virtual hosts are used for load balancing between two or more OracleAS Single Sign-On servers. They also provide an extra layer of security.

virtual host name

In an Oracle Application Server Cold Failover Cluster (Identity Management), the host name corresponding to a particular virtual IP address.

virtual IP address

In an Oracle Application Server Cold Failover Cluster (Identity Management), 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.

wait time

The time between the submission of the request and initiation of the response.

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.

Wallet Manager

See [Oracle Wallet Manager](#).

Web service

A Web service is application or business logic that is accessible using standard Internet protocols, such as [HTTP](#), [XML](#), and [SOAP](#). Web Services combine the best aspects of component-based development and the World Wide Web. Like components, Web Services represent black-box functionality that can be used and reused without regard to how the service is implemented.

Web Services Description Language (WSDL)

Web Services Description Language (WSDL) is the standard format for describing a Web service using [XML](#). A WSDL definition describes how to access a Web service and what operations it will perform.

WSDL

See [Web Services Description Language \(WSDL\)](#).

WS-Federation

Web Services Federation Language (WS-Federation) is a specification developed by Microsoft, IBM, BEA, VeriSign, and RSA Security. It defines mechanisms to allow [federation](#) between entities using different or like mechanisms by allowing and brokering trust of identities, attributes, and authentication between participating [Web services](#).

See also: [Liberty Alliance](#).

X.500

X.500 is a standard from the International Telecommunication Union (ITU) that defines how global directories should be structured. X.500 directories are hierarchical with different levels for each category of information, such as country, state, and city.

X.509

X.509 is the most widely used standard for defining digital certificates. A standard from the International Telecommunication Union (ITU), for hierarchical directories with authentication services, used in many **public key infrastructure (PKI)** implementations.

XML

Extensible Markup Language (XML) is a specification developed by the World Wide Web Consortium (W3C). XML is a pared-down version of Standard Generalized Mark-Up Language (SGML), designed especially for Web documents. XML is a metalanguage (a way to define tag sets) that allows developers to define their own customized markup language for many classes of documents.

XML canonicalization (C14N)

This is a process by which two logically equivalent XML documents can be resolved to the same physical representation. This has significance for digital signatures because a signature can only verify against the same physical representation of the data against which it was originally computed. For more information, see the W3C's XML Canonicalization specification.

Index

A

- abstract object classes, 2-13
 - superclasses of, 8-3
 - top, 2-12
- access
 - exclusionary, 14-13
 - granting
 - by using command-line tools, 14-34
 - by using Oracle Directory Manager, 14-13
 - entry-level, by using command-line tools, 14-36
 - entry-level, by using Oracle Directory Manager, 14-24
 - kinds, 14-8
 - level requirements for LDAP operations, 14-10
 - object, 14-6
 - operations, 14-8
 - rights, setting by using Oracle Directory Manager, 14-17, 14-22
 - selecting, by DN, 14-37
 - subject, 14-7
 - unspecified, 14-10, 14-23
 - violation event, 10-9
- access control
 - and authorization, 2-15
 - conceptual discussion, 12-2
 - default, 17-3
 - defined, 2-15
 - directive format. See ACI directive format
 - management constructs, 14-2
 - managing, 14-1
 - by using command-line tools, 14-34
 - by using Oracle Directory Manager, 14-13
 - overview, 1-7
 - policies
 - conflicting, 14-2
 - inheriting, 14-2
 - policy administration, overview, 14-1
 - prescriptive, 14-2
 - setting, by using wildcards, 14-36
- access control information (ACI)
 - attributes, 12-2
 - components, 14-6
 - directives, format
 - items
 - format, C-1
 - syntax, C-1
 - more than one for the same subject, 14-12
 - object of directives, 14-6
 - subject of directives, 14-7
- access control lists (ACLs), 2-7, 12-2
 - directives, within entries, 14-2
 - evaluation
 - for groups, 14-13
 - precedence rules, 14-11
 - for groups, 14-13
 - how it works, 14-10
 - modification, 10-9
 - precedence
 - rules, 14-11
 - within subtrees, 14-2
- Access Control Management pane, in Oracle Directory Manager, A-4
- access control policy points (ACPs), 14-2, 14-15
 - adding
 - by using ldapmodify, 14-36
 - by using Oracle Directory Manager, 4-4, 14-15
 - by using the ACP Creation Wizard of Oracle Directory Manager, 14-18
 - administering, by using Oracle Directory Manager, 4-7
 - configuring display of, in Oracle Directory Manager, 14-14
 - creating by using ACP Creation Wizard, 14-18
 - Creation Wizard, 14-18
 - defined
 - groups, 14-3
 - multiple, 14-2
 - viewing, 14-15
 - by using Oracle Directory Manager, 14-15
 - viewing, by using Oracle Directory Manager, 14-15
- accounts
 - enabling and disabling
 - by using command-line tools, 15-7
 - by using Oracle Internet Directory Self-Service Console, 15-8
 - unlocking
 - by using command-line tools, 15-7
 - by using Oracle Internet Directory Self-Service Console, 15-8

- realm administrator, 19-11
 - super user, 15-2
- ACI. See access control information (ACI)
- ACL. See access control lists (ACLs)
- ACP groups, 14-3
- ACP. See access control policy points (ACPs)
- ACPs. See access control policy points (ACPs)
- active server instances
 - modifying configuration set entries in, 5-3
 - viewing, 5-3, 5-11
- added_object_constraint filter, 14-35
- added-object-constraint, in access control, 14-8
- administration tools, 6-7
 - Catalog Management Tool (catalog.sh), 4-16
 - command-line, 1-6, 4-14
 - Human Intervention Queue Manipulation Tool, 4-18
 - ldapadd, 6-7
 - ldapdelete, 6-7
 - ldapmodify, 6-8
 - OID Database Password Utility (oidpasswd), 4-19
 - OID Database Statistics Tool (oidstats.sql), 4-18
 - OID Migration Tool, 4-18
 - OID Reconciliation Tool, 4-18
 - Oracle Directory Manager, 4-1
 - Replication Environment Management Tool, 4-18
- Advanced Replication
 - configuring
 - by using Replication Management Tool, 25-7
- agents
 - log file location, 3-4
- alias entries
 - adding, 5-13
 - dereferencing, 5-11, 5-12
 - messages, 5-15
 - modifying, 5-15
 - searching directory with, 5-14
- ANALYZE function of DBMS_STATS package, 21-2
- anonymous authentication, 12-3, A-2
- anonymous login, A-2
- Application Server Control
 - starting directory server instance, 10-17
 - stopping directory server instance, 10-18
 - viewing user logon session information, 10-18
- application-specific repositories
 - migrating data from, 23-8
- Apply button, in Oracle Directory Manager, 4-3
- architecture
 - Oracle Internet Directory, 1-5, 2-1
 - Oracle Internet Directory Server Manageability framework, 10-14
- ASR Agreement tab page, in Oracle Directory Manager, A-11
- attribute options, 2-11
 - adding
 - by using ldapmodify, 6-9
 - by using Oracle Directory Manager, 6-6
 - conceptual discussion, 2-11
 - deleting by using Oracle Directory Manager, 6-7,
- 6-9
 - language codes, 2-11
 - managing
 - by using command line tools, 6-9
 - by using Oracle Directory Manager, 6-6
 - modifying by using Oracle Directory Manager, 6-6
 - searching for by using ldapsearch, 6-9
- attribute uniqueness
 - about, 7-1
 - constraint entries, 7-1
 - entries
 - location of, 7-5
 - known limitations, 7-8
 - managing, 7-5
 - managing by using command-line tools, 7-6
 - managing, by using Oracle Directory Manager, 7-5
 - rules for creating, 7-2
- attributes
 - adding, 8-8
 - by using ldapmodify, 8-13
 - by using Oracle Directory Manager, 8-10, 8-11
 - guidelines for, 8-8
- as DNs, 6-3
- as metadata in schema, 8-1
- attribute options, 6-9
 - adding by using ldapmodify, 6-9
 - adding by using Oracle Directory Manager, 6-6
 - conceptual discussion, 2-11
 - deleting by using Oracle Directory Manager, 6-7, 6-9
 - managing by using command line tools, 6-9
 - managing by using Oracle Directory Manager, 6-6
 - modifying by using Oracle Directory Manager, 6-6
- base schema
 - deleting, 8-9
 - modifying, 8-9
- commonName, 2-10
- creating by using Oracle Directory Manager, 4-4
- deleting, 8-9
 - guidelines for, 8-9
- determined by object classes, 8-2
- ditcontentrule, 8-16
- dropping indexes, 8-12
- extending number of
 - by using auxiliary object classes, 8-16
 - by using content rules, 8-16
 - for existing entries, 8-16
 - prior to creating entries, 8-15
- for a specific entry
 - viewing by using Oracle Directory Manager, 6-3
- for which data exists
 - indexing, 8-15
- for which no directory data exists
 - indexing, 8-14

- in base schema, 8-8
- in top, 2-13
- indexed
 - viewing, 8-12
- indexes, created by bulkload, 6-11
- indexing, 8-15
 - by using Catalog Management tool
 - by using command-line tools, 8-14
 - by using Oracle Directory Manager, 8-12
 - when you create them, 8-12
- information, kinds of, 2-9
- inheritance of, 8-2
- jpegPhotos, 2-10, 6-8
- kinds of information in, 2-9
- labeledURI, 9-2, 9-8, 9-10
- making available for searches, 8-12
- managing, 8-8
 - by using command-line tools, 8-13
 - by using Oracle Directory Manager, 8-8, 8-9
 - overview, 8-8
- managing by using command-line tools, 8-13
- mandatory, 2-12, 6-5, 8-2
 - in a user entry, 23-10
- matching rules, 2-11
- modifying
 - by using ldapmodify, 6-8
 - by using ldapmodifymt, 6-8
 - by using Oracle Directory Manager, 6-6, 8-11
 - guidelines for, 8-9
 - rules for, 8-9
 - using ldapmodify, 8-13
- multivalued, 2-10, 14-3
 - converting to single-valued, 8-9
- null values in, 8-2
- objectclass, 10-8
- objects associated with an ACI, 14-6
- operational, 5-7
- optional, 2-12, 8-2
- options, 2-11
 - language codes., 2-11
- orclauditlevel, 10-10
- orclauditmessage, 10-8
- orclauditoc, 10-8
- orcleventtime, 10-8
- orcleventtype, 10-8
- orclopresult, 10-8
- orclsequence, 10-8, 10-9
- orcluserdn, 10-8
- organization, 2-10
- organizationalUnitName, 2-10
- redefining mandatory, 8-3
- ref, 6-12
- removing from object classes, 8-4
- rules
 - for adding, 8-8
 - for deleting, 8-9
 - for modifying, 8-9
- searching for, by using Oracle Directory Manager, 8-9
 - single-valued, 2-10
 - converting to multivalued, 8-9
 - skewed, optimizing searches for, 21-9
 - sn, 2-10
 - specifying as mandatory or optional, 8-2
 - surname, 2-10
 - syntax, 2-11
 - modifying, 8-9
 - syntax type
 - selecting, 8-20
 - syntaxes
 - cannot modify, 8-9
 - selecting, 8-20
 - system operational, 5-7
 - types, 2-9
 - usercertificate, G-1
 - values, 2-9
 - viewing, 6-3
- Attributes tab page, in Oracle Directory Manager, A-16
- audit level, 10-9
 - modifying, 10-11
 - setting, 10-10
 - by using ldapmodify, 10-11
 - by using Oracle Directory Manager, 10-10
- audit log, 10-7
- container object, 10-12
- default configuration, 10-7
- entries
 - in the DIT, position of, 10-9
 - position in DIT
 - searching, 10-8
 - searching for, 10-11
 - searching for by using ldapsearch, 10-12
 - searching for by using Oracle Directory Manager, 10-11
 - structure, 10-8
 - viewing, 10-7
- events
 - access violation, 10-9
 - ACL modification, 10-9
 - add, 10-9
 - adding, 10-9
 - bind, 10-9
 - deleting, 10-10
 - DSE modification, 10-9
 - modify, 10-10
 - modifyDN, 10-10
 - modifying, 10-10
 - replication login, 10-9
 - schema element, add/replace, 10-9
 - schema element, delete, 10-9
 - selected, 10-10
 - super user login, 10-9
 - user password modification, 10-9
- garbage collector, 22-2
- purging, 10-12
- queries, 10-7
- sample, 10-9
- structure of entries, 10-8

- using, 10-7
- auditable events, 10-9
- auditing selected events, 10-10
- authenticated access, by using SSL, 1-7
- authentication, 12-3
 - anonymous, 12-3, A-2
 - certificate, 12-4
 - conceptual discussion, 12-3
 - defined, 2-15
 - direct
 - options, 12-3
 - external, 12-6, 28-1
 - SASL, 12-4
 - in a typical directory operation, 2-7
 - indirect, 12-4
 - through a RADIUS server, 12-4
 - native, 28-1
 - Oracle directory replication server, 24-14
 - password-based, 12-3, A-2
 - PKI, 12-1
 - SASL, 12-3
 - SASL mechanism
 - external authentication, 12-4
 - MD5 Digest, 12-3
 - simple, 1-7, 12-3, A-2
 - Simple Authentication and Security Layer (SASL), 12-3
 - SSL
 - defined, 12-3
 - for Oracle Directory Manager, A-3
 - no, A-3
 - server only, A-3
 - three levels, 1-7
 - through a middle tier, 12-4
- Authentication Choice list, in Oracle Directory Manager, A-4
- Authentication Services Group, 17-12
- authorization, 2-15, 12-2
- automated resolution of conflicts, 24-20
- auxiliary object classes, 2-13, 8-4
 - extending number of attributes by using, 8-16
- average latency, 21-1

B

- backup and recovery strategies, failover, 18-4
- backup and restore, 11-1
- base schema
 - attributes, 8-8
 - deleting, 8-9
 - modifying, 8-9
 - object classes
 - modifying, 8-4
- base search, 6-2
- batching line-mode commands, 8-7
- Begins With filter, in Oracle Directory Manager, A-15
- bind event, 10-9
- bind mode, 14-8
- binding, 2-7

- BSTAT/ESTAT scripts, 21-6
- buffer caches, size, 21-6
- bulk loading failure, 6-11
- bulkdelete, 4-17, 6-12
 - and Globalization Support, D-9
- bulkload, 4-17, 6-11
 - and Globalization Support, D-8
 - check mode, performing on LDIF files, 23-8
 - creating indexes, 6-11
 - .dat files, 6-11
 - generating input files, 6-11
 - load option, 6-11
 - log file location, 3-4
- bulkmodify, 4-17
 - and Globalization Support, D-9
- By Whom tab page, in Oracle Directory Manager, A-4

C

- C API, 2-7
- cache, entry, 21-8
- cache, metadata, 2-4
- caching
 - client-side referral, 6-14
- Cancel button, in Oracle Directory Manager, 4-3
- capacity planning, 18-5, 18-6, 20-1
 - I/O subsystem, 20-5
 - network requirements, 20-10
 - overview, 20-1
- catalog entry, 2-5
- Catalog Management Tool (catalog.sh), 4-16, 8-15
 - log file location, 3-4
- cataloged attributes
 - orcleventype, 10-8
 - orcluserdn, 10-8
- catalog.sh. See Catalog Management tool.
- certificate authentication, 12-4
- change log
 - purging, in multimaster replication, 22-5
- change log garbage collection
 - troubleshooting, J-21
- Change Log window, in Oracle Directory Manager, A-13
- change logs, 2-18, 24-5
 - and directory replication, 24-15
 - change number-based purging, 22-5
 - garbage collector, 22-2
 - in replication, 1-7, 24-15, 24-19
 - purging, 22-5
 - methods, 22-5
 - time-based purging, 22-5
- change number-based purging, 22-5
- change retry count, setting, A-11
- cipher suites
 - SSL, 13-1
 - SSL, supported, 13-1
 - SSL_RSA_WITH_3DES_EDE_CBC_SHA, 13-1
 - SSL_RSA_WITH_NULL_MD5, 13-1
 - SSL_RSA_WITH_NULL_SHA, 13-1

- SSL_RSA_WITH_RC4_128_SHA, 13-1
- client-side referral caching, how it works, 6-14
- cn attribute, 2-10
- cn=replication namecontext, 24-11
- command line tools
 - described, 4-14
- command-line tools, 1-6
 - adding configuration set entries, 2-6, 6-7
 - Catalog Management Tool, 8-12
 - comparing attribute values, 6-7
 - for managing entries, 6-7
 - indexing, 8-12, 8-15
 - ldapadd, 6-7
 - ldapaddmt, 6-7
 - ldapdelete, 6-7
 - ldapmodify, 6-8
 - managing
 - attributes, 8-13
 - entries, 6-7
 - modifying configuration set entries, 6-8
 - overview, 4-14
 - setting Globalization Support, D-5
- common entry, defined, 2-5
- Common Group Attributes Group, 17-14
- Common User Attributes Group, 17-14
- commonName attribute, 2-10
- comparing
 - attribute values, 6-7
 - entries, 6-7
 - two objects, 4-4
- component deployment and administration
 - delegation, 17-8
- Components
 - Process Control, 4-8
- components
 - of a directory server, 2-2
 - of Oracle Internet Directory, 1-6
- concurrent database connections, 21-7
- configsets, 2-6
- configuration parameters
 - modifying, 2-6
 - Oracle directory replication server
 - location, 25-35
- configuration set entries, 2-6
 - adding, 2-6, 5-1, 5-5
 - by using command line tools, 6-7
 - by using command-line tools, 2-6
 - by using Oracle Directory Manager, 5-3
 - changing, 5-6
 - deleting, 5-1
 - by using ldapmodify, 5-6
 - by using Oracle Directory Manager, 5-3, 5-5
 - for replication server, 25-35
 - LDIF files, 5-5
 - managing, 4-20, 5-1
 - by using command-line tools, 5-5
 - by using Oracle Directory Manager, 5-3
 - preliminary considerations, 5-2
 - modifying, 2-6, 5-1, 5-2
 - by using ldapmodify, 5-6
 - by using Oracle Directory Manager, 5-3, 5-5
 - in an active server instance, 5-3
 - multiple, 13-3
 - overriding user-specified, J-14
 - SSL parameters in, 13-2
 - using different, 5-2
 - viewing, 5-3
- configuration set location, A-22
- Configuration Sets General tab page, in Oracle Directory Manager, A-21
- conflict resolution, in replication, 24-19
- conflicting access control policies, 14-2
 - precedence, rules for resolving, 14-2
- conflicts, replication
 - automated resolution of, 24-20
 - manual resolution of, 25-18
 - resolution, 14-11, 24-19
 - resolving manually, 25-18
 - typical causes of, 24-20
- CONNECT BY assertions, in dynamic groups, 9-3
- Connect/Disconnect button in Oracle Directory Manager, 4-5
- connecting
 - to a directory server, 4-2, 4-20
 - in a typical directory operation, 2-7
 - to additional directory servers, 4-5
 - to multiple directory servers, 4-5
- connection
 - pooling, 1-7
- connections, LDAP, specifying maximum idle time
 - for, 5-11
- constraints, object classes, 2-13
- consumers
 - defined, 2-17, 24-2
- containment
 - of groups, planning, 19-3
 - of users, planning, 19-3
- content access items, 14-26
 - of an existing ACP, 14-23
- Content Rule dialog box, in Oracle Directory Manager, A-19
- content rules
 - defined, 8-16
 - defined as values of ditcontentrule attribute, 8-16
 - extending number of attributes by using, 8-16
 - managing
 - by using command-line tools, 8-18
 - by using Oracle Directory Manager, 8-18
 - rules for creating and modifying, 8-16
 - schema enforcement when using, 8-17
- control, access, 1-7, 14-1
- controls
 - definition, 1-4
- converting
 - auxiliary object classes, 8-4
 - directory data to LDIF, 6-12
 - structural object classes, 8-4
- CPUs
 - configuration, 20-11
 - in capacity planning, 20-1

- power required for various deployment scenarios, 18-7
- processing power, 20-11
- requirements, 20-11
 - detailed calculations, 20-12
 - in capacity planning, 20-11
- tuning, 21-3
- tuning for Oracle foreground processes, 21-4
- usage, 18-8
- usage tuning, 21-3
- when to tune, 21-3
- Create button, in Oracle Directory Manager, 4-5
- Create Entry menu item, in Oracle Directory Manager, 4-4
- Create Like
 - adding entries using templates, 6-4
 - button, in Oracle Directory Manager, 4-5, 6-4
 - operation, by using Oracle Directory Manager, 4-4
- createTimestamp attribute, 2-10, 23-7
 - optional in top, 2-13
- creatorsName attribute, 2-10, 23-7
 - optional attribute in top, 2-13
- critical events
 - in Oracle Internet Directory Server Manageability framework, 10-16
 - levels, 10-16

D

- .dat files, generated by bulkload, 6-11
- data integrity, 2-15, 12-1
- data migration process, 23-2
- data privacy, 2-15, 12-1
 - by using SSL, 1-7
- data, updating by using Oracle Directory Manager, 4-5
- database
 - block buffers parameter, 21-7
 - block size parameter, 21-7
 - cache size, 18-7
 - connections, 2-4
 - concurrent, 21-7
 - pooling, 1-7
 - dedicated for directory, 2-3
 - password, changing, 5-11
 - queries, optimization of, 21-9
 - server, 1-5
 - server error, J-2
 - tuning, 21-6
- DB_BLOCK_BUFFERS, 21-6
- DBMS_STATS package, 21-2
- debug
 - log files, viewing, J-15
- debug dimension, 10-5
- debug logging
 - levels, 10-4
 - about, 10-1
 - setting, 10-4
 - setting by using OID Control Utility, 10-4

- setting by using Oracle Directory Manager, 10-4
- levels, setting
 - by using OID Control Utility, 10-4
 - by using Oracle Directory Manager, 10-4
- log files, viewing, 10-4
- debugging the external authentication plug-in, 28-3
- debugging, limiting to specific operations, 10-5
- default
 - identity management realm, 2-25, 19-10
- default configuration
 - access controls, 17-3
- default knowledge references (referrals)
 - configuring, 6-13
- default knowledge references (referrals),
 - configuring, 6-13
- default port, 4-2
- Delegated Administration Services
 - defined, 2-21
 - overview, 2-26
- delegation
 - component deployment and administration, 17-8
 - how it works, 17-1
 - in an Oracle Application Server environment, 17-2
 - of privileges for user and group management, 17-3
- Delete button, in Oracle Directory Manager, 4-5
- deployment
 - considerations, 18-1
 - CPU power, 18-7
 - failover, 18-4
 - replication, 18-3
 - tuning, 18-8
 - partitioning, 18-3
- deployment considerations
 - metadirectory, 18-5
- dereferencing alias entries, 5-12
- DES40 encryption, 12-1
- descriptions of object classes, A-14, A-16
- Digest
 - MD5, 12-3
- directories
 - access control, 1-7, 14-1
 - application, migrating data from, 23-8
 - as read-focused, 1-2
 - backup and restore, 11-1
 - contrasted to relational databases, 1-2
 - database listener, 25-9
 - defined, 1-1
 - distributed, 2-16
 - existing, default directory structure, 23-1
 - expanding role of, 1-1, 18-1
 - location-independent, 1-2
 - multimaster replication groups (DRGs)
 - installing, 25-3
 - online
 - expanding role of, 1-1
 - partitioned, 2-18
 - password, changing, 5-9

- planning structure of, 19-3
- read-focused, 1-2
- replication groups (DRGs), 24-16, 25-3
 - and replication agreements, 24-16
 - configuring, 25-3
- schema
 - managing, 8-1
 - overview, 8-1
- small
 - backing up and restoring, 11-1
 - special purpose, 1-2
- directory information tree (DIT), 2-7
 - audit log entries in
 - browsing, 6-2
 - default, 19-10
 - planning for identity management, 19-1
- directory integration and provisioning server
 - log file location, 3-4
- directory metadata
 - defined, 2-4
- Directory Replication Group, 25-2
- directory replication server, 1-6, 2-3
 - authentication, 24-14
 - configuration set entries, 25-35
 - log file location, 3-4
- directory schema, 8-1
 - defined, 2-4
 - managing, 8-1
- directory servers, 1-6, 2-3
 - adding, A-2
 - as both suppliers and consumers, 24-19
 - changing parameters in an active instance, 5-3
 - configuration set entries, 5-1
 - connecting to, 4-2, 4-5, 4-20, A-2
 - by using Oracle Directory Manager, 4-5
 - in a typical directory operation, 2-7
 - connecting to additional, 4-5
 - connecting to one on a different host, A-2
 - connecting to, by using Oracle Directory Manager, 4-4
 - disconnecting from, using Oracle Directory Manager, 4-6
 - disconnecting, by using Oracle Directory Manager, 4-4, 4-6
 - discovery by using the Domain Name System (DNS), 5-17
 - in multi-master replication, 24-19
 - in replicated environment, 24-19
 - locating in a distributed environment, 5-16
 - log file location, 3-4
 - modifying, A-2
 - modifying configuration set entries, 5-6
 - multimaster replication between, 1-7
 - parameters
 - configuring, 4-20
 - configuring by using command-line tools, 4-20
 - processes, 2-4
 - multiple, 2-4
 - restarting, 5-3
 - restarting, by using the Application Server Control, 10-18
 - shared server, 1-7
 - specifying host, A-2
 - starting
 - by using Application Server Control, 10-17
 - with different configuration, J-14
 - static discovery by using ldap.ora, 5-16
 - stopping
 - by using Application Server Control, 10-18
 - terminating, 4-20
 - user logon session information
 - viewing by using Application Server Control, 10-18
 - using different configuration set entries, 5-2
 - viewing information, 5-11
 - directory usage patterns, learning, 20-2
 - DirectoryReplicationGroupDSAs, 25-39
 - Disconnect
 - button, in Oracle Directory Manager, 4-4
 - menu item, in Oracle Directory Manager, 4-4
 - disconnecting from directory servers, 4-6
 - disk space requirements, 20-5
 - detailed calculations of, 20-6
 - estimating, 20-5
 - disk tuning, 21-6
 - disk usage, 18-8
 - displaying
 - a directory entry, 6-1
 - a subtree, 6-1
 - distinguished names, 2-7
 - as attributes, 6-3
 - components of
 - format
 - modifying
 - by using command line tools, 6-8
 - distributed directories, 2-16, 2-18
 - locating directory servers in, 5-16
 - partitioned, 2-16
 - partitions and replicas, 18-2
 - partitions, replicas, and high availability, 18-2
 - replicated, 2-16
 - DIT. See directory information tree (DIT)
 - ditcontentrule attribute, 8-16
 - DNs. See distinguished names.
 - Domain Name System (DNS)
 - registering a directory server with, 5-18
 - server discovery by using, 5-17
 - DRG, 25-2
 - Drop Index
 - button, 4-5
 - menu item, 4-4
 - DSE modification event, 10-9
 - duration of a search, specifying, 6-2, 10-12
 - dynamic directory server discovery, 5-16, 5-17
 - dynamic groups, 9-2
 - entries
 - managing by using command-line tools, 9-10
 - managing by using Oracle Directory Manager, 9-9

- schema elements for creating, 9-2
- dynamic password verifiers
 - troubleshooting, J-22

E

- E argument in Globalization Support, D-5

Edit

- button, in Oracle Directory Manager, 4-5
- menu item, in Oracle Directory Manager, 4-4

encryption

- DES40, 12-1
- levels available in Oracle Internet Directory, 12-1
- password, 12-6
- passwords
 - UNIX crypt, 16-3, 16-4
 - RC4_40, 12-2

- Encryption Choice list, in Oracle Directory Manager, A-4

- Ends With filter, in Oracle Directory Manager, A-15

- entity component, in access control, 14-7

entries

- adding
 - by copying an existing entry, 6-4
 - by using ldapadd, 6-7
 - by using ldapaddmt, 6-7
 - by using Oracle Directory Manager, 6-3
 - concurrently, 6-7
 - mandatory attributes, 6-4
 - optional attributes, 6-4
 - requires write access to parents, 6-3
- alias, dereferencing, 5-11
- attributes, viewing, 6-3
- audit log, 10-7
 - searching, 10-8
- command-line tools for managing, 6-7
- comparing, by using ldapcompare, 6-7
- conceptual discussion, 2-7
- configuration set, 2-6
- creating by using Oracle Directory Manager, 4-4
- deleting
 - by using ldapdelete, 6-7
 - large numbers, 6-12
- displaying, 6-1
- distinguished names of, 2-7
- garbage collector, 22-4
- group, 2-10
- inheriting attributes, 8-2
- loading, 8-2
- locating by using distinguished names
- managing, 6-1
 - by using bulk tools, 6-9
 - by using command line tools, 6-7
 - by using Oracle Directory Manager, 4-7, 6-1
- managing by using command-line tools, 6-7
- many, modifying, 6-12
- modifying
 - by using Oracle Directory Manager, 6-5
- naming, 2-7
- objects associated with an ACI, 14-6

- parent, 8-2
- replication naming context container, 24-11
- restricting the kinds users can add, 14-17, 14-19, 14-22, 14-25, 14-35

- root of search, 6-1

searching

- base level, 6-2
- by using Oracle Directory Manager, 6-1
- one-level, 6-2
- specifying search depth, 6-2
- subtree level, 6-2

- selecting by DN, 14-37

- selecting superclass, 6-3

- specific, granting access to, A-4

static group

- modifying, by using ldapmodify, 9-7, 9-11

- statistics collector, 22-4

- superclasses, selecting, 6-3

user

- adding, by using ldapadd, 6-8

- adding, by using Oracle Directory Manager, 6-4

- modifying, 6-8

- modifying, by using ldapmodify, 6-8

- modifying, by using Oracle Directory Manager, 6-5

with attribute options

- adding by using ldapmodify, 6-9

- adding by using Oracle Directory Manager, 6-6

- deleting by using Oracle Directory Manager, 6-7, 6-9

- managing by using command line tools, 6-9

- managing by using Oracle Directory Manager, 6-6

- modifying by using Oracle Directory Manager, 6-6

- searching for by using ldapsearch, 6-9

entry

- caching, 21-8

- enabling, A-23

- catalog, defined, 2-5

- common, defined, 2-5

- password policy, defined, 2-6

- password verifier, defined, 2-6

- plug-in, defined, 2-6

- entry-level access, granting by using Oracle Directory Manager, 14-24

- environment variables, NLS_LANG, D-3

- error messages, J-4

- additional, J-4

- administration, J-2

- database server, J-2

- directory server, due to schema modifications, J-2

- installation, J-1

- ORA-1562, J-2

- password policies, J-7

- returned from Oracle directory server, J-2

- standard, J-2

- events, auditable, 10-9

- Exact Match filter, in Oracle Directory Manager, A-15, A-27
- exclusionary access to objects, granting, 14-13
- existing ACPs and their ACI directives, modifying, 14-21
- Exit menu item, in Oracle Directory Manager, 4-4
- explicit hierarchies, 9-4
- extensibility, in LDAP Version 3, 1-4
- extensibleObject object class, 6-12
- external authentication, 12-6
 - contrasted with native authentication, 28-1
 - defined, 28-1
 - plug-in, 28-1
 - debugging, 28-3
 - installing, 28-2, 28-4
 - installing, configuring, and enabling, 28-2
 - SASL authentication mechanism, 12-4
- external repository, storing security credentials in, 28-1

F

- failover, 1-7
 - considerations in deployment, 18-4
- failure to apply changes, 24-19
- failure tolerance, and replication, 18-4
- fan-out replication, 2-17, 24-2, 24-21
 - groups, 2-17, 24-2, 24-5
 - in conjunction with multimaster replication groups, 24-6
 - LDAP-based, 2-17
 - process, 24-21
- features, new, i-xxxiii
 - in Oracle Internet Directory, Release 3.0.1, i-xliv
 - release 10g (10.1.2), i-xxxiii
 - release 10g (9.0.4), i-xxxiv
 - release 2.1.1, i-xlvi
 - release 3.0.1, i-xliv
 - release 9.0.2, i-xl
- File menu, in Oracle Directory Manager, 4-4
- filters
 - Begins With, A-15
 - Ends With, A-15
 - Exact Match, A-15, A-27
 - Greater or Equal, A-15, A-27
 - in attribute searches, 8-10
 - in searches, 2-7, 8-5
 - in Oracle Directory Manager, 8-5
 - Less or Equal, A-15, A-27
 - not null, A-15
 - Present, Oracle Directory Manager, A-27
- Find Attributes button, in Oracle Directory Manager, 8-10
- Find Objects button, in Oracle Directory Manager, 4-5, 8-5
- formats, of distinguished names
- full replication, 2-17, 24-1
- function calls, tracing, 10-5

G

- garbage collection
 - framework
 - about, 22-1
 - components of, 22-1
 - how it works, 22-3
 - in replication, 22-5
 - plug-in, 22-1
- Garbage Collector window, in Oracle Directory Manager, A-6
- garbage collectors
 - audit log, 22-2
 - change log, 22-2
 - definition, 22-2
 - entries for, 22-4
 - general statistics, 22-2
 - health statistics, 22-2
 - managing, 22-6
 - modifying
 - by using command-line tools, 22-6
 - by using Oracle Directory Manager, 22-6
 - predefined, 22-2
 - security and refresh events, 22-2
 - system resource events, 22-3
 - tombstone, 22-3
- general statistics garbage collector, 22-2
- Globalization Support, 2-15
 - bulkdelete, D-9
 - bulkload, D-8
 - bulkmodify, D-9
 - command-line tools, D-5
 - Java clients, 2-16
 - ldapadd, D-6
 - ldapaddmt, D-6
 - ldapbind, D-6
 - ldapcompare, D-7
 - ldapdelete, D-6
 - ldapmoddn, D-7
 - ldapmodify, D-6
 - ldapmodifymt, D-6
 - ldapsearch, D-6
 - ldifwrite, D-8
 - managing, D-1
 - settings for Oracle Internet Directory, D-3
 - using with Bulk Tools, D-7
 - with bulkdelete, D-9
 - with bulkload, D-8
 - with bulkmodify, D-9
 - with command-line tools, D-5
 - with LDIF Files, D-4
 - with ldifwrite, D-8
- Greater or Equal filter, in Oracle Directory Manager, A-15, A-27
- group entries, 2-10
 - adding, 6-5, 9-6
 - creating
 - by using Oracle Directory Manager, 9-6, 9-9
- groupOfNames object class, 9-6, 9-9
- groupOfUniqueNames object class, 9-6, 9-9
- groups

- ACL evaluation for, 14-13
- ACP, 14-3
- dynamic, 9-2
 - managing by using command-line tools, 9-10
 - managing by using Oracle Directory Manager, 9-9
 - schema elements for creating, 9-2
- dynamic and static, administration of, 9-1
- granting access rights to, 14-4
- hierarchical, 9-4
- membership
 - how directory server computes, 14-4
- names and containment, planning, 19-3
- privilege, 14-3
 - defined, 2-5
- static, 9-1
 - managing by using command-line tools, 9-7
 - managing by using Oracle Directory Manager, 9-6
 - schema elements for creating, 9-1
 - when to use static or dynamic, 9-5
- guest users
 - definition, 5-9
 - managing, 5-9
 - by using ldapmodify, 5-10
 - by using Oracle Directory Manager, 5-9
 - user name and password, 5-9
- guidelines
 - for adding attributes, 8-8
 - for adding object classes, 8-3
 - for deleting attributes, 8-9
 - for deleting object classes, 8-4
 - for modifying attributes, 8-9
 - for modifying object classes, 8-3

H

- hashing
 - passwords to the directory, 16-1
 - protection
 - MD4, 16-2
- hashing algorithm
 - default, 16-2
 - userPassword, 16-2
- health statistics garbage collector, 22-2
- Help
 - button, in Oracle Directory Manager, 4-5
 - menu item, in Oracle Directory Manager, 4-4
- hierarchical groups, 9-4
- hierarchies
 - explicit, 9-4
 - implicit, 9-4
- high availability, 1-7, 18-2, 18-4
 - considerations, 18-4
- Human Intervention Queue Manipulation Tool, 4-18, 25-19

I

- identity management, 19-10

- defined, 2-23
- Oracle Identity Management infrastructure, 19-1
- planning DIT for, 19-1
- policies, 2-25
- realms
 - configuring, 19-12
 - customizing, 19-12
 - default, 2-25
 - defined, 2-25
 - entry in default DIT, 19-9
 - implementation in Oracle Internet Directory, 19-9
 - in enterprise deployments, 19-7
 - in hosted deployments, 19-8
 - multiple in enterprise deployments, 19-8
 - planning, 19-5
 - single in enterprise deployments, 19-7
 - realm-specific Oracle Context, 19-10
- identity management realms, 2-25, 19-7
 - creating additional, 19-18
 - multiple, 19-8
 - single, 19-7
- idle time, specifying maximum for LDAP connections, 5-11
- IETF
 - LDAP approval
- implicit hierarchies, 9-4
- indexed attributes
 - locations, A-23
 - orcleventype, 10-8
 - orcluserdn, 10-8
 - viewing, 8-12
- indexes
 - created by bulkload, 6-11
 - dropping from attributes, 8-12, 10-8
 - by using Oracle Directory Manager, 8-12
- inheritance, 2-12
 - and access control policies, 14-2
 - from superclasses, 8-2
- input file, creating, 5-6
- installation errors, J-1
- installation types
 - in multimaster replication group
 - installation, 25-3
- insufficient memory, 21-6
- intelligent client failover, 18-4
- intelligent network level failover, 18-4
- intermediate template file
 - in migration from application-specific repositories, 23-8
- internationalization, and LDAP, D-1
- Internet Engineering Task Force (IETF). See IETF.
- introduction to LDAP and Oracle Internet Directory, 1-1
- I/O subsystem, 20-5
 - in capacity planning, 20-1, 20-5
 - requirements, 20-4
 - sizing, 20-5
 - throughput, maximizing, 20-5
- iostat utility, 21-2

J

Java clients, Globalization Support and, 2-16
Java Native Interface, 2-7
jpegPhoto attribute, 2-10, 6-8

K

knowledge references, 2-19, 2-20, 18-2, 18-3
 configuring, 6-12
 default
 configuring, 6-13
 defined, 2-19
 managing, 6-12
 overview, 2-19
 restricting permissions for managing, 2-20
 smart
 configuring, 6-12
 superior, 2-19

L

labeledURI attribute, 9-2, 9-8, 9-10
language codes, as attribute options, 2-11
latency, average, 21-1
LDAP
 add or modify performance, J-9
 and internationalization, 2-15
 and simplified directory management, 1-3
 attributes, common, 2-10
 extensibility, 1-4
 IETF approval, 1-3
 search performance, J-9
 security, 1-4
 server instances, 2-3
 servers, 2-3
 managing, 5-1
 multithreaded, 1-7
 Version 3, 1-3
LDAP connections, specifying maximum idle time for, 5-11
LDAP Data Interchange Format (LDIF), 4-14
LDAP dispatcher
 log file location, 3-4
LDAP replica states, H-1
ldapadd, 6-7
 and Globalization Support, D-6
ldapaddmt, 6-7
 and Globalization Support, D-6
LDAP-based partial replication
 determining what is to be replicated, 25-32
LDAP-based replica
 configuring, 25-22
 deleting, 25-31
LDAP-based replication, 2-17, 24-2
 agreements, 24-8
 configuring, 25-20
 replica states, H-1
ldapbind
 and Globalization Support, D-6
ldapbind operation, 12-3

ldapcompare, 6-7
 and Globalization Support, D-7
LDAP-compliant directories, migrating data from, 23-2
ldapdelete, 6-7
 and Globalization Support, D-6
ldapmoddn, 6-7
 and Globalization Support, D-7
ldapmodify, 6-8
 adding ACPs, 14-36
 adding attributes, 8-13
 adding entry-level ACIs
 adding object classes, 8-7
 and Globalization Support, D-6
 changing audit level, 10-11
 modifying attributes, 8-13
 modifying object classes, 8-7
ldapmodifymt, 6-8
 and Globalization Support, D-6
ldap.ora, 5-16
 server discovery by using, 5-16
ldapssearch, 6-8
 and Globalization Support, D-6
 querying audit log, 10-7
LDIF
 converting directory data to, 6-12
 files
 creating, 5-6
 for adding configuration set entries, 5-5
 importing by using bulkload, 6-10
 importing, by using bulkload, 6-10
 referencing in commands, 5-7
 removing proprietary data from in migration, 23-7
 using, 4-14
ldifmigrator, 4-18
ldifwrite, 4-17
 and Globalization Support, D-8
Less or Equal filter, A-15, A-27
line-mode commands, batching, 8-7
listener, for directory database, 2-3, 2-4
 restarting, 25-9
 stopping, 25-9
listener.ora, 25-9
load balancing
 and replication, 18-4
-load option, in bulkload, 6-11
location-independence, of directories, 1-2
log files
 debug, viewing, 10-4, J-15
 locations, 3-4
logging
 for garbage collectors, enabling and disabling, 22-7
login
 anonymous, A-2
 super user, A-2
 user, A-2
loose consistency model of replication, 18-3
LSNRCTL utility, 25-9

M

- managing
 - directory schema, 8-1
- mandatory attributes, 2-12, 8-2
 - adding to existing object classes, 8-4
 - adding to object classes in use, 6-5
 - entering values for, 6-4
 - in a user entry, 23-10
 - in object classes, A-15, A-16
 - redefining, 8-3
- manual resolution of conflicts, 25-18
- matching rules
 - as metadata in schema, 8-1
 - attribute, 2-11
 - cannot add to subSchemaSubentry, 8-1
 - stored in schema, 8-1
- Matching Rules tab page, in Oracle Directory Manager, A-18
- MD4, 16-3, 23-7
- MD5, 16-3, 23-7
 - for password encryption, 16-2, 16-4
- MD5 Digest, SASL authentication mechanism, 12-3
- member attribute, 9-6, 9-9
- memory
 - in capacity planning, 20-1
 - insufficient, 21-6
 - physical, 20-9
 - required, 18-7
 - requirements in capacity planning, 20-9
 - tuning, 21-5
 - usage, 18-8
 - virtual, 20-9
- menu bar, Oracle Directory Manager, 4-3
- metadata
 - cache, 2-4
 - directory, defined, 2-4
 - stored in schema, 8-1
- metadirectory
 - deployment considerations, 18-5
- middle tier
 - using proxy user with, 5-9, 12-4
- migrating data, 23-2, 23-6
 - from other LDAP directories, 23-2
 - from other LDAP-compliant directories, 23-1
- migration
 - from application-specific repositories, 23-8
 - intermediate template file, 23-8
 - from other LDAP directories, 23-2
- modifiersName attribute, 2-10, 23-7
 - optional in top, 2-13
- modifyDN, audit log event, 10-10
- modifyTimestamp attribute, 2-10, 23-7
 - optional in top, 2-13
- monitoring servers, 10-12
- mpstat utility, 21-2
- multimaster flag
 - toggling, 25-12
- multimaster replication, 1-7, 2-17, 18-3, 18-4, 24-2
 - architecture, 24-17
 - conflict resolution, 24-19

- groups, 24-4
 - in conjunction with fan-out replication groups, 24-6
 - installation types, 25-3
 - installing, 25-3
 - on the consumer side, 24-18
 - on the supplier side, 24-17
- multiple configuration set entries, 13-3
- multiple server processes, 2-4
- multithreaded command-line tools
 - ldapaddmt, 6-7
- multithreaded LDAP servers, 1-7
- multivalued attributes, 2-10
 - converting to single-valued, 8-9
- member, 9-6, 9-9
- orclEntryLevelACL, 14-3

N

- names
 - of groups, planning, 19-3
 - of users, planning, 19-3
- names, of object classes, A-14, A-16
- naming contexts, 2-14
 - backing up and restoring, 11-1
 - definition, 2-14
 - discovering, 2-14
 - in partitioned directories, 2-18
 - managing, 5-8
 - publishing, 2-14, 5-8
 - by using ldapmodify, 5-8
 - by using Oracle Directory Manager, 5-8
 - searching for published, 5-8
 - subordinate, 2-19
- namingContexts attribute, 5-8
 - multivalued, 5-8
- native authentication
 - contrasted with external authentication, 28-1
 - defined, 28-1
- navigator pane, in Oracle Directory Manager, 4-3
- network
 - bandwidth, 20-10
 - capacity planning, 20-10
 - connectivity, in capacity planning, 20-1
 - requirements, 20-10
- New Attribute Type Advanced tab page, in Oracle Directory Manager, A-18
- New Attribute Type General tab page, in Oracle Directory Manager, A-17
- New Constraint dialog box, in Oracle Directory Manager, A-5
- New Content Rule dialog box, in Oracle Directory Manager, A-19
- new features, i-xxxiii
 - release 10g (10.1.2), i-xxxiii
 - release 10g (9.0.4), i-xxxiv
 - release 2.1.1, i-xlvi
 - release 3.0.1, i-xxliv
 - release 9.0.2, i-xl
- New Plug-in dialog box, in Oracle Directory

- Manager, A-9
- new syntaxes, adding, 2-11
- NLS_LANG environment variable, D-3
 - setting, D-3
 - in the client environment, D-7
 - settings, D-3
- no SSL authentication option, A-3
- nodes, Oracle Internet Directory, 2-2
- non-default port, running on, 4-2
- not null filter, in Oracle Directory Manager, A-15
- null values, in attributes, 8-2
- number of worker threads used in change log processing, modifying, 25-37

O

- o attribute, 2-10
- O3LOGON algorithm, 16-4
- object
 - adding, by using Oracle Directory Manager, 4-4
- object class types
 - structural, 2-13
- object classes, 2-12
 - adding, 8-2
 - by using command-line tools, 8-7
 - by using Oracle Directory Manager, 8-5
 - as metadata in schema, 8-1
 - assigning to entries, 8-2
 - auxiliary, 2-13
 - converting auxiliary, 8-4
 - creating, by using Oracle Directory Manager, 4-4
 - defining, 8-15
 - deleting
 - by using Oracle Directory Manager, 8-6
 - from base schema, 8-16
 - not in base schema, 8-4
 - explosion, 8-3
 - extensibleObject, 6-12
 - groupOfNames, 9-6, 9-9
 - guidelines
 - for adding, 8-3
 - for deleting, 8-16
 - for modifying, 8-3
 - in the base schema, modifying, 8-4
 - managing
 - by using command-line tools, 8-7
 - by using Oracle Directory Manager, 8-1
 - modifying, 8-3
 - by using command-line tools, 8-7
 - by using Oracle Directory Manager, 8-6
 - orclacpgroup, 14-3
 - orclauditoc, 10-8
 - orclprivilegegroup, 2-5
 - and dynamic groups, 9-4
 - redefining mandatory attributes in, 8-3
 - referral, 6-12
 - removing attributes from, 8-4
 - removing superclasses from, 8-4
 - rules, 2-13
 - searching for, 8-4

- searching for, by using Oracle Directory Manager, 8-4
- structural, 2-13
- structural, converting, 8-4
- subclasses, 2-12
 - defining, 8-15
- superclasses, 2-12
- top, 2-12
- types, 2-12
 - abstract, 2-13
 - auxiliary, 2-13
 - structural, 2-12
- types of, 2-12
- unique name of, 8-3
- unique object identifier, 8-3
- viewing, 8-5
- viewing properties, 8-5
- object identifiers, of object classes, A-14, A-16
- objectclass attribute, 10-8
- objects
 - adding, by using a template, 4-5
 - adding, by using Oracle Directory Manager, 4-5
 - comparing, 4-4
 - modifying
 - by using ldapmodify, 6-8
 - by using Oracle Directory Manager, 4-4, 4-5
 - of ACI directives, 14-6
 - removing
 - by using Oracle Directory Manager, 4-4, 4-5
 - searching for
 - by using Oracle Directory Manager, 4-4, 4-5
 - searching for, by using Oracle Directory Manager, 4-5
- OCI. See Oracle Call Interface.
- ODS_PROCESS Table, 4-13
- OID Control Utility
 - restart command, 5-3
 - viewing debug log files, 10-4, J-15
- OID Database Password Utility, 5-11
- OID Database Password Utility (oidpasswd), 4-19
- OID Database Statistics Tool, 4-18
- OID Migration Tool, 4-18
- OID Monitor, 2-3, 4-16
 - log file location, 3-4
- OID Password Utility, 3-3
- OID Reconciliation Tool, 4-18, 25-19
- OIDCTL, 4-14
- oidctl
 - viewing debug log files, 10-4, J-15
- oidctl. See OID Control Utility
- oidexaup.sql
 - contents of, 28-4
 - for installing external authentication plug-in, 28-2
- OIDEXTAUTH PL/SQL package for external authentication, 28-2
- oidldapd
 - log file location, 3-4
- OIDMON, 4-13
- oidstats.sql, 4-18
- OLTS_ATTRSTORE tablespace, 20-9

- OLTS_CT_STORE tablespace, 20-9
- OLTS_DEFAULT tablespace, 20-9
- one-level search, 6-2
- one-way authentication, SSL, A-3
- online administration tool. See Oracle Directory Manager
- online directories, 1-1
- open cursors parameter, 21-7
- OPEN_CURSORS, 21-7
- OpenLDAP Community, 0-xxxxi
- operation debug dimension, 10-5
- operational attributes, 5-7
 - ACI, 12-2
- operation-based plug-ins, 26-2
- Operations menu item, in Oracle Directory Manager, 4-4
- operations, limiting debugging to specific, 10-5
- OPMN, 4-8
- optional attributes, 2-12, 8-2
 - adding to pre-defined object classes, 8-15
 - entering values for, 6-4
 - in object classes, A-15, A-16
- options, attribute, 2-11
- Oracle, 2-7
- Oracle Advanced Security, use of Oracle Internet Directory, 1-8
- Oracle Application Server Administrators Group, 17-9
- Oracle Application Server Certificate Authority
 - part of Oracle Identity Management, 1-4
- Oracle Application Server Portal, use of Oracle Internet Directory, 1-8
- Oracle Application Server Single Sign-On
 - use of Oracle Internet Directory, 1-8
- Oracle background processes, 21-7
- Oracle Call Interface
- Oracle Collaboration Suite, use of Oracle Internet Directory, 1-8
- Oracle components
 - privileges for administering, 17-3
- Oracle components, use of Oracle Internet Directory, 1-7
- Oracle Context
 - root, 19-9
- Oracle Context Administrators Group, 17-13
- Oracle data servers
 - changing password to, 5-11
 - error messages, J-2
- Oracle Database Advanced Replication, 24-16, 25-9
 - agreements, 24-8
 - configuring, 25-7, 25-9
 - by using Replication Management Tool, 25-7
 - for directory replication, 25-9
 - directory replication based on, 2-17, 24-2
 - installing, 25-7
 - setting up, 25-7
- Oracle Delegated Administration Services
 - overview, 2-26
 - part of Oracle Identity Management, 1-4
- Oracle Directory Integration and Provisioning
 - platform, 1-9
 - part of Oracle Identity Management, 1-4
 - what it is, 2-22, 18-5
- Oracle Directory Manager, 6-2
 - Access Control Management pane, A-4
 - adding
 - ACPs, 14-15
 - attributes, 8-10
 - configuration set entries, 5-3
 - entries, 6-3
 - group entries, 6-5, 9-6
 - object classes, 8-5
 - objects, 4-4
 - Apply button vs. OK button, 4-3
 - ASR Agreement tab page, A-11
 - Attributes tab page, A-16
 - attributes, searching for, 8-9
 - Authentication Choice list, A-4
 - By Whom tab page, A-4
 - Cancel button, 4-3
 - Change Log window, A-13
 - Configuration Sets General tab page, A-21
 - connecting to a directory server, 4-4, 4-5
 - Content Rule dialog box, A-19
 - create access control policy point menu, 4-4
 - Create button, 4-5
 - Create Entry menu item, 4-4
 - Create Like button, 4-5, 6-4
 - Create Like operation, 4-4
 - creating an attribute, 4-4
 - creating object classes, 4-4
 - defined
 - Delete button, 4-5
 - deleting
 - configuration set entries, 5-3
 - objects, 4-5
 - disconnecting from a directory server, 4-4
 - displaying help navigator, 4-4
 - Edit button, 4-5
 - Edit menu, 4-4
 - Encryption Choice list, A-4
 - Ends With filter, A-15
 - entries management, 4-7
 - Exact Match filter, A-15, A-27
 - Exit menu item, 4-4
 - File menu, 4-4
 - Find Attributes button, 8-10
 - Find Objects button, 4-5, 8-5
 - Garbage Collector window, A-6
 - granting access, 14-13
 - Greater or Equal filter, A-15, A-27
 - Help button, 4-5
 - Help menu item, 4-4
 - launching, 4-2
 - Less or Equal filter, A-15, A-27
 - managing
 - ACPs, 4-7
 - configuration set entries, 5-3
 - entries, 4-7
 - object classes, 8-1

- Matching Rules tab page, A-18
- menu bar, 4-3
- modifying
 - configuration set entries, 2-6, 5-3
 - entries, 6-5
 - object classes, 8-6
 - objects, 4-4, 4-5
 - replication agreements, 25-39
- navigating, 4-3
- New Attribute Type Advanced tab page, A-18
- New Attribute Type General tab page, A-17
- New Constraint dialog box, A-5
- New Content Rule dialog box, A-19
- New Plug-in dialog box, A-9
- not null filter, A-15
- on UNIX, starting, 4-2
- on Windows, starting, 4-2
- Operations menu, 4-4
- overview, 4-1, 4-4
- Password Policies Account Lockout tab page, A-8
- Password Policies General tab page, A-6
- Password Policies IP Lockout tab page, A-8
- Password Policies Password Syntax tab page, A-8
- Password Verifier Profile dialog box, A-8
- Present filter, A-27
- Query Optimization tab page, A-26
- Refresh button, 4-5
- Refresh Entry button, 4-5
- Refresh Subtree Entries button, 4-5
- removing objects, 4-4
- Replica Agreements tab page, A-12
- Replica Naming Context tab page, A-13
- Replica Node General tab page, A-12
- Replication Server Configuration Set General tab page, A-11
- Revert button, 4-3
- root of search, 6-1
- running, 4-2
- schema administration, 4-8
- search criteria bar, 6-2, 10-12
- search filters, 8-5
- searching
 - entries, 6-1
 - for an object, 4-5
 - for attributes, 8-9
- selecting attribute syntax type, 8-20
- SSL Settings tab page, A-28
- starting, 4-2
 - on UNIX, 4-2
 - on Windows, 4-2
- Synchronization Execution tab page, A-30
- Synchronization General tab page, A-29
- Synchronization Mapping tab page, A-31
- Synchronization Status tab page, A-31
- system operation attributes displayed, A-21
- System Passwords tab page, A-25
- tear-off menu item, 4-4
- toolbar, 4-5
- updating, 4-4
 - subtree entry data, 4-5
 - View menu, 4-4
 - viewing attributes, 6-3
- Oracle directory replication server
 - authentication, 24-14
 - component of Oracle Internet Directory, 1-6
 - component of Oracle Internet Directory node, 2-3
 - configuration parameters, location, 25-35
 - starting, 25-11
 - uses LDAP to communicate to directory server, 2-3
- Oracle directory server instance, 2-3
- Oracle directory server instances, 1-6, 2-3
 - managing, 5-1
 - starting, 25-11
 - stopping, 3-1
- Oracle directory version field, in Oracle Directory Manager, A-22
- Oracle foreground processes
 - tuning CPU for, 21-4
- Oracle Globalization Support, 2-15
- Oracle Identity Management, 2-23
 - and Oracle Internet Directory, 1-4, 19-1
 - components, 2-24
 - delegation in, 17-1
 - group information, 19-4
 - in application deployments, 1-4
 - infrastructure, 2-23
 - what it does, 2-23
 - management policies, 2-25
 - objects, 19-9
 - planning, 19-2
 - realms, planning, 19-5
 - user information, 19-4, 19-11
- Oracle Internet Directory
 - advantages of, 1-7
 - and Oracle Identity Management, 1-4
 - architecture, 1-5, 2-1
 - components, 1-6
 - how Oracle components use it, 1-7
 - multiple installations on same host, 18-5
 - nodes, 2-2
 - used by Oracle Advanced Security, 1-8
 - used by Oracle Application Server Single Sign-On, 1-8
- Oracle Internet Directory Self-Service Console, 2-21
 - in indirect authentication of end users, 12-4
- Oracle Internet Directory Server Manageability
 - architecture and components, 10-14
 - capabilities, 10-13
 - configuring, 10-15
 - framework, 10-12
 - configuring critical events, 10-16
 - location of configuration information, 10-15
 - managing, 10-17
- Oracle Net Services, 2-3, 2-7
 - preparing for replication, 25-7
 - use of Oracle Internet Directory, 1-8
- Oracle Real Application Clusters, i-xliv
- OracleApplication Server Single Sign-On
 - part of Oracle Identity Management, 1-4

- Oraclei
 - database, 2-3
- orclACI, 14-2
 - access to, 14-2
 - optional attribute in top, 2-13
- orclacpgroup object class, 14-3
- orclAgreementID, 25-39
- orclauditlevel attribute, 10-10
- orclauditlevel operational attribute, 10-7
- orclauditmessage attribute, 10-8
- orclauditoc attributes, 10-8
- orclauditoc object class, 10-8
- orclChangeRetryCount, 25-36
- orcldebugflag, 10-4
- orclDirReplGroupDSAs, 25-41
- orclEntryLevelACI, 14-2, 14-3
 - optional attribute in top, 2-13
- orcleventtime attribute, 10-8
- orcleventtype attribute, 10-8
- orclxcludedattributes, 24-11
- orclxcludednamingcontexts, 24-11
- orclGuid
 - optional attribute in top, 2-13
- orclguname attribute, 5-10
- orclgupassword attribute, 5-10
- orclincludednamingcontexts, 24-11
- ORCLLM algorithm, 16-4
- ORCLMAXCC, 21-3
- orclmaxcc, 2-4
- ORCLNT algorithm, 16-4
- orclopresult attribute, 10-8
- orclpkimatchingrule, 12-4
- orclprivilegegroup object class, 2-5
 - and dynamic groups, 9-4
- orclprname attribute, 5-10
- orclprpassword attribute, 5-10
- orclrevpwd attribute, 16-2
- orclsequence attribute, 10-8, 10-9
- ORCLSERVERPROCS, 21-3
- orclskewedattribute attribute, 21-9
- orclsunname attribute, 5-10
- orclsupassword attribute, 5-10
- orcluniqueattrname, 7-2
- orcluniqueenable, 7-2
- orcluniqueobjectclass, 7-2
- orcluniquescope, 7-2
- orcluniquesubtree, 7-2
- orcluserdn attribute, 10-8
- orclUserV2 attribute, 23-11
- ORCLWEBDAV algorithm, 16-4
- organization attribute, 2-10
- organizationalUnitName, 2-10
- overall throughput, 21-1

P

- paging, 20-9
- parameters
 - configuration, for Oracle directory replication server, 25-35

- dependent on Oracle directory server
 - configuration, 21-7
 - for an active instance, modifying, 13-3
 - in an active server instance
 - modifying, 5-3
 - replication agreement, 25-39
 - required for tuning, 21-7
 - SGA, 21-8
- partial replication, 2-17, 24-1
- partitioning, 2-16, 2-18
 - deployment considerations, 18-3
- partitions, 18-2
- password policies, 12-6
 - about, 15-1
 - conceptual discussion, 12-6
 - default, 15-1
 - definition, 15-1
 - entry
 - defined, 2-6
 - error messages, J-7
 - establishing, 15-4
 - for realms
 - modifying by using command-line tools, 15-7
 - viewing by using command-line tools, 15-7
 - management, 2-15
 - managing by using command-line tools, 15-6
 - plug-in, 27-1
 - how it works, 27-1
 - possible problems, J-7
 - realms, managing by using command-line tools, 15-7
 - realm-specific
 - modifying by using Oracle Directory Manager, 15-6
 - viewing by using Oracle Directory Manager, 15-5
 - setting by using command-line tools, 15-6
 - setting by using Oracle Directory Manager, 15-4
 - setting, by using command-line tools, 15-6
 - troubleshooting, J-7
 - verification of, 15-3
- Password Policies Account Lockout tab page, in Oracle Directory Manager, A-8
- Password Policies General tab page, in Oracle Directory Manager, A-6
- Password Policies IP Lockout tab page, in Oracle Directory Manager, A-8
- Password Policies Password Syntax tab page, in Oracle Directory Manager, A-8
- password verifier entry, defined, 2-6
- Password Verifier Profile dialog box, in Oracle Directory Manager, A-8
- password verifiers
 - default, for Oracle components, 16-7
- password wallets
 - troubleshooting, J-22
- password-based authentication, 12-3, A-2
- passwords
 - database, 5-11
 - for guest users, 5-9

- for proxy users, 5-9
- for shell tools, 6-11
- for SSL wallets, A-3
- for super user, 5-9
- for super users, 5-9
- forcing changes by using command-line tools, 15-8
- integrity
 - MD4, 16-2
- policies, 12-6
 - setting by using command-line tools, 15-6
 - setting by using Oracle Directory Manager, 15-4
- protection, 2-15, 12-6
 - changing by using ldapmodify, 16-3
 - changing by using Oracle Directory Manager, 16-3
 - changing scheme, 16-1
 - default verifiers for Oracle components, 16-7
 - managing by using ldapmodify, 16-3
 - managing by using Oracle Directory Manager, 16-3
 - MD5, 16-2, 16-4
 - O3LOGON, 16-4
 - ORCLLM, 16-4
 - ORCLNT, 16-4
 - ORCLWEBDAV, 16-4
 - SASL/MD5, 16-4
 - setting by using Oracle Directory Manager, A-24
 - SHA, 16-2, 16-4
 - UNIX Crypt, 16-3, 16-4
- to a directory, changing, 5-9
- to Oracle data servers, changing, 5-11
- peer-to-peer replication, 2-17, 24-2
- performance
 - add or modify, J-9
 - by using orclEntryLevelACI, 14-2
 - metrics, 21-1
 - replication and, 18-4
 - search, J-9
 - troubleshooting, J-9
 - tuning, tools for, 21-2
- permissions, 2-15, 12-2
 - granting
 - by using command-line tools, 14-34
 - by using Oracle Directory Manager, 14-13
- physical distribution, partitions and replicas, 18-2
- physical memory, 20-9
- PKI authentication, 12-1
- plug-ins
 - adding, 26-4, 26-5
 - deleting, 26-6
 - entry, 2-6
 - external authentication, 28-1
 - for password policies, 27-1
 - framework, 26-1
 - garbage collection, 22-1
 - modifying, 26-5
 - operation-based, 26-2
 - password policy
 - how it works, 27-1
 - post-operation, 26-2
 - pre-operation, 26-2
 - registering
 - by using command-line tools, 26-4
 - by using Oracle Directory Manager, 26-3
 - when-operation, 26-3
 - point-to-point replication, 2-17, 24-2
 - policies
 - identity management, 2-25
 - pooling, connection, 1-7
 - port, A-3
 - default, 4-2
 - precedence
 - at the attribute level, 14-12
 - at the entry level, 14-11
 - rules
 - ACL evaluation, 14-11
 - in conflicting access policies, 14-2
 - prescriptive access control, 14-2
 - Present filter, Oracle Directory Manager, A-27
 - privacy, data, 2-15, 12-1
 - by using SSL, 1-7
 - privilege groups, 14-3
 - associated with orclPrivilegeGroup object class, 14-4
 - defined, 2-5
 - privileges, 2-15, 12-2
 - privileges for user and group management
 - delegation of, 17-3
 - Process control of Oracle Internet Directory
 - components, 4-8
 - process instance location, A-24
 - processes, 2-3
 - Oracle background, 21-7
 - processing power of CPU, 20-11
 - processor affinity on SMP systems, 21-5
 - proxy users, 12-4
 - definition, 5-9
 - managing, 5-9
 - by using ldapmodify, 5-10
 - by using Oracle Directory Manager, 5-9
 - user name and password, 5-9
 - public key infrastructure, 12-1
 - pwdPolicy object class, 15-4

Q

- queries, database
 - optimizing, 21-9
- query entry return limit, A-24
- Query Optimization tab page, in Oracle Directory Manager, A-26
- querying
 - audit log, 10-7
 - critical events, 10-7

R

- RC4_40 encryption, 12-2
- RDNs. See relative distinguished names (RDNs)
- realms, 19-7
 - identity management
 - configuring, 19-12
 - customizing, 19-12
 - default, 2-25, 19-10
 - defined, 2-25
 - implementation in Oracle Internet Directory, 19-9
 - in enterprise deployments, 19-7
 - in hosted deployments, 19-8
 - multiple in enterprise deployments, 19-8
 - planning, 19-5
 - single in enterprise, 19-7
 - realm-specific Oracle Context, 19-10
 - recovery features, in Oracle, 1-7
 - redefining mandatory attributes, 8-3
 - redo log buffers parameter, 21-8
 - redundancy
 - and failover, 18-2
 - ref attribute, 6-12
 - referral caching, client-side, 6-14
 - how it works, 6-14
 - referral object class, 6-12
 - referrals, 2-19
 - client-side referral caching, 6-14
 - defined, 2-19
 - kinds, 2-20
 - Refresh button, in Oracle Directory Manager, 4-5
 - Refresh Entry button, in Oracle Directory Manager, 4-5
 - Refresh Entry menu item, 4-4
 - Refresh Subtree Entries button, in Oracle Directory Manager, 4-5
 - Refresh Subtree Entries menu item, 4-4
 - relational databases contrasted to directories, 1-2
 - relative distinguished names (RDNs)
 - displaying for each entry, 6-1
 - modifying
 - by using command line tools, 6-8
 - modifying, by using ldapmoddn, 6-7
 - remtool, 25-9
 - replica
 - subentry, 24-10
 - Replica Agreements tab page, in Oracle Directory Manager, A-12
 - Replica Naming Context tab page, in Oracle Directory Manager, A-13
 - Replica Node General tab page, in Oracle Directory Manager, A-12
 - replica state, H-1
 - replicas, 2-17, 18-2, 24-1
 - in deployment, 18-2
 - replicated directories, conceptual discussion, 2-16
 - replication, 2-17, 3-5, 24-19
 - adding a new entry to a consumer, F-1
 - adding a new node for, 25-13
 - agreement entry, 24-10
 - agreement parameters, 25-39
 - modifying, 25-39, 25-40
 - viewing and modifying, 25-39, 25-40
 - agreements, 2-17, 24-1, 24-8, 25-39, A-24
 - adding nodes to, 25-40
 - configuring, 25-39
 - example of, 24-12
 - LDAP-based, 24-8
 - Oracle Database Advanced Replication, 24-8
 - and SSL, 24-15
 - architecture, 24-17
 - authentication, 24-14
 - change conflicts
 - monitoring, 25-18
 - change logs, 1-7, 24-19
 - change logs in, 24-15
 - comparison of full and partial, 24-3
 - configuration parameters
 - modifying, 25-36
 - configuring, 25-35
 - Oracle Database Advanced Replication, 25-9
 - sqlnet.ora, 25-8
 - tnsnames.ora, 25-8
 - conflicts
 - levels of occurrence, 24-19
 - resolving manually, 25-18
 - typical causes of, 24-20
 - considerations, 18-3
 - deleting a node, 25-17
 - deleting an entry, F-2
 - failure tolerance, 18-4
 - fan-out, 2-17, 24-2, 24-5, 24-21
 - process, 24-21
 - full, 2-17, 24-1, 24-2
 - groups, 24-3
 - fan-out, 24-5
 - multimaster, 2-17, 24-4
 - single-master, 24-4
 - in deployment, 18-3
 - installing and configuring, 25-3
 - LDAP-based, 2-17, 24-2
 - configuring, 25-20, 25-22
 - deleting, 25-31
 - determining what is to be replicated, 25-32
 - installing and configuring, 25-20
 - load balancing, 18-4
 - log location, A-24
 - login events, 10-9
 - loose consistency model, 18-3
 - managing, 25-1
 - managing naming contexts and attributes, 24-27
 - modifying a DN, F-4
 - modifying an RDN, F-3
 - multimaster, 1-7, 2-17, 18-3, 24-2
 - architecture, 24-17
 - conflict resolution, 24-19
 - installing and configuring, 25-1
 - on the consumer side, 24-18
 - on the supplier side, 24-17
 - multimaster with fan-out, 24-6

- multimaster, single-master, fan-out, 24-6
- naming context container entry, 24-11
- naming contexts
 - included and excluded, 24-7
- nodes
 - adding, 25-13
 - deleting, 25-17
- Oracle Database Advanced Replication, 24-2
- Oracle Database Advanced Replication-based, 2-17
- Oracle Databased Advanced filtering, 24-22
- overview, 24-1
- partial, 2-17, 24-1, 24-2
 - filtering, 24-22
 - optimization, 24-28
- peer-to-peer, 2-17, 24-2
- point-to-point, 2-17, 24-2
- preparing the Oracle Net Services environment for, 25-7
- process, F-1, F-2, F-3, F-4
 - on the consumer side, 24-18
 - on the supplier side, 24-17
- reasons to implement, 18-4
- retries
 - applying changes, 24-19
- security, 24-14
- server
 - log file location, 3-4
 - single-master, 2-17
 - specifying number of worker threads, A-11
 - status location, A-24
- Replication Environment Management Tool, 4-18
- Replication Management Tool, 25-7
- Replication Server Configuration Set General tab page, in Oracle Directory Manager, A-11
- resource access information, 2-26
- resource information, 2-26
 - location in DIT, 2-26
- resource type information, 2-26
- Revert button, in Oracle Directory Manager, 4-3
- Root DSE entry
 - defined, 2-5
- root of search
 - entering, 6-1
 - selecting, 6-2
- root Oracle Context, 19-9

S

- SASL
 - clients enabled with
 - Digest-MD5 authentication to directory server, 12-7
 - external authentication, 12-7
 - SASL/MD5, for generating password verifier, 16-4
 - scalability, of Oracle Internet Directory, 1-7
 - schema
 - adding and changing object classes (online), 8-2
 - administration, 8-1

- by using Oracle Directory Manager, 4-8
- definition location, A-24
- definitions in subSchemaSubentry, 8-1
- directory, defined, 2-4
- elements
 - add/replace event, 10-9
 - delete event, 10-9
 - for orclACI, C-1
 - for orclEntryLevelACI, C-2
 - objects, administering by using Oracle Directory Manager, 4-8
- scripts, batched line-mode commands, 8-7
- search
 - and compare operations, 2-11
 - criteria bar, in Oracle Directory Manager, 6-2, 10-12
 - depth, specifying, 6-2
 - results, specifying maximum number of entries returned, 6-2, 10-11
- Search ACPs
 - button, 4-5
 - menu item, 4-4
- searches
 - configuring, 5-11
 - for ACPs when using Oracle Directory Manager, 14-14
 - configuring display and duration of, 4-6
 - duration, 10-12
 - specifying maximum number of entries returned, 6-2, 10-11
 - using filters, 8-5
- secure
 - port 636, 13-2, 13-3
- Secure Hash Algorithm (SHA), 16-3, A-24
- secure mode
 - running server instances in, 13-3
- security, 1-7, 2-14
 - credentials, stored in an external repository, 28-1
 - for different clients, 13-3
 - in LDAP Version 3, 1-4
 - in replication, 24-14
 - SSL parameters for different clients, 13-3
 - within Oracle Internet Directory environment, 2-15
- Security Administrators Group, 17-12
- security and refresh events garbage collector, 22-2
- selected audit log events, 10-10
- server
 - instances
 - running, 4-2
 - running in secure mode, 13-3
 - mode, A-24
 - operation time limit, A-24
- servers
 - configuring
 - by using input files, 6-7
 - monitoring, 10-12
- servers. See also directory servers, directory replication servers, or directory integration and provisioning servers

- Service Registry, 2-21
- Service to Service Authentication, 2-21
- SESSIONS parameter, 21-7
- setup process (ldaprepl.sh)
 - log file location, 3-5
- SHA, 16-3, 23-7, A-24
 - for password encryption, 16-2, 16-4
- shared pool size, 21-6
 - parameter, 21-7
- shared server, 21-8
- simple authentication, 1-7, 12-3
- Simple Authentication and Security Layer (SASL)
 - authentication, 12-3
 - clients enabled with
 - Digest-MD5 authentication to directory, 12-7
 - external authentication, 12-7
 - how it works, 12-7
 - in LDAP Version 3, 1-4
 - single-master replication groups, 24-4
 - single-valued attributes, 2-10
 - converting to multivalued, 8-9
 - size
 - of database cache, 18-7
 - sizing, 18-5, 18-6
 - considerations in deployment, 18-6
 - I/O subsystem, 20-5
 - tablespaces, 20-6
 - skewed attributes, 21-9
 - smart knowledge references (referrals)
 - configuring, 6-12
 - sn attribute, 2-10
 - sort area parameter, 21-8
 - special purpose directories, 1-2
 - SPECint_rate95 baseline, 20-11
 - sponsor node, 25-14
 - sqlnet.ora, configuring for replication, 25-8
 - SRV records
 - OID-specific format for, 5-18
 - standard format for, 5-19
- SSL, 13-4, A-3
 - authenticated access, 1-7
 - authentication
 - for Oracle Directory Manager, A-3
 - one-way, A-3
 - server only, A-3
 - cipher suites, 13-1
 - SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA, 13-2
 - SSL_DH_anon_EXPORT_WITH_RC4_40_MD5, 13-2
 - SSL_DH_anon_WITH_3DES_EDE_CBC_SHA, 13-1
 - SSL_DH_anon_WITH_DES_CBC_SHA, 13-1
 - SSL_DH_anon_WITH_RC4_128_MD5, 13-1
 - SSL_RSA_EXPORT_WITH_DES40_CBC_SHA, 13-1, 13-2
 - SSL_RSA_EXPORT_WITH_RC4_40_MD5, 13-1
 - SSL_RSA_WITH_DES_CBC_SHA, 13-1
 - SSL_RSA_WITH_NULL_SHA, 13-1
 - SSL_RSA_WITH_RC4_128_MD5, 13-1
 - supported in Oracle Internet Directory, 13-1
 - client scenarios, 13-2
 - configuration parameters, 13-2
 - modifying, 13-3
 - configuring, 13-2, A-2
 - data privacy, 1-7
 - enabling, 13-2
 - enabling Oracle Directory Manager to use, A-3
 - handshake, 13-1
 - issues specific to this release, 13-2
 - managing, 13-1
 - no authentication, A-3
 - parameters, 13-2
 - configuring by using command-line tools, 13-4
 - configuring by using Oracle Directory Manager, 13-3
 - password to user wallet, A-3
 - port 636, 13-3
 - replication and, 24-15
 - strong authentication, 12-1
 - Version 2, 13-2
 - Version 3, 13-2
 - SSL Settings tab page, in Oracle Directory Manager, A-28
 - SSL setup
 - troubleshooting, J-20
 - SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA, 13-2
 - Starting directory components, 4-8
 - start-server commands, 5-1
 - static directory server discovery, 5-16
 - static groups, 9-1
 - entries
 - managing by using command-line tools, 9-7
 - managing by using Oracle Directory Manager, 9-6
 - modifying by using ldapmodify, 9-7, 9-11
 - schema elements for creating, 9-1
 - statistics collector
 - entry for, 22-4
 - store-and-forward transport, in Oracle Database Advanced Replicaton, 24-16
 - striping, 21-6
 - strong authentication, 12-3
 - structural access items, 14-25
 - structural object class type, 2-12, 2-13
 - structural object classes, 2-13
 - converting, 8-4
 - structure rules, not enforced by Oracle Internet Directory, 2-13
 - structure, audit log entries, 10-8
 - subclasses, 2-12
 - subentries, definition, 8-1
 - subordinate naming contexts, 2-19
 - subSchemaSubentry
 - adding object classes to, 8-1
 - holding schema definitions, 8-1
 - modifying, 8-1

- subtree entry data, updating by using Oracle Directory Manager, 4-5
- subtree level search, 6-2
- subtrees
 - displaying, 6-1
- super users
 - definition, 5-9
 - logging in as, A-2
 - login events, 10-9
 - managing, 5-9
 - by using ldapmodify, 5-10
 - by using Oracle Directory Manager, 5-9
 - user name and password, 5-9
- superclass selector, 6-3
- superclasses, 2-12
 - and inheritance, 8-2
 - of object classes, A-14, A-16
- superior knowledge references (referrals), 2-19
- suppliers
 - defined, 2-17, 24-2
- surname attribute, 2-10
- Symmetric Multi-Processor (SMP) systems, 21-5
- Synchronization Execution tab page, in Oracle Directory Manager, A-30
- Synchronization General tab page, in Oracle Directory Manager, A-29
- Synchronization Mapping tab page, in Oracle Directory Manager, A-31
- Synchronization Status tab page, in Oracle Directory Manager, A-31
- syntax
 - attribute, 2-11
 - stored in schema, 8-1
- syntaxes
 - cannot add to subSchemaSubentry, 8-1
 - new, adding, 2-11
 - viewing
 - by using by using ldapsearch, 8-20
 - by using Oracle Directory Manager, 8-20
- System Global Area (SGA), 21-5
 - parameters, 21-8
 - sizing, 21-5
 - tuning for Oracle, 21-5
 - tuning parameters, 21-8
- system operation attributes
 - displayed in Oracle Directory Manager, A-21
- system operational attributes, 5-7
 - setting, 5-7
 - by using ldapmodify, 5-7
 - by using Oracle Directory Manager, 5-7
 - viewing, 5-7
- System Passwords tab page, in Oracle Directory Manager, A-25
- system resource events garbage collector, 22-3
- SYSTEM tablespace, 20-9

T

- tablespaces, 20-6
 - OLTS_ATTRSTORE, 20-9

- OLTS_CT_STORE, 20-9
- OLTS_DEFAULT, 20-9
 - sizing, 20-6
 - SYSTEM, 20-9
- TCP/IP Problems, J-1
- tear-off, in Oracle Directory Manager, 4-4
- templates, creating entries from, 6-4
- throughput, 20-5
 - overall, 21-1
- time-based change log purging, 22-5
- tnsnames.ora
 - configuring for replication, 25-8
- tombstone garbage collector, 22-3
- tools
 - for tuning, 21-2
- top object class, 2-12, 2-13
 - optional attributes in, 2-13
- top utility, 21-2
- tracing function calls, 10-5
- tree view
 - browsing, 6-2
 - selecting root of search, 6-2
- troubleshooting, J-1
 - change log garbage collection, J-21
 - directory server instance startup, J-10
 - dynamic password verifiers, J-22
 - password policies, J-7
 - password wallets, J-22
 - performance, J-9
 - SSL setup, J-20
 - typical problems in Oracle Internet Directory, J-1
- Trusted Application Administrators Group, 17-10
- tunables, database, 21-6
- tuning, 18-5, 21-1
 - considerations, 18-8
 - CPU for Oracle foreground processes, 21-4
 - CPU for Oracle Internet Directory processes, 21-3
 - CPU usage, 21-3
 - deployment considerations, 18-8
 - disk, 21-6
 - memory, 21-5
 - overview, 21-1
 - SGA parameters, 21-8
 - System Global Area (SGA) for Oracle, 21-5
 - tools, 21-2
- types
 - of attributes, 2-9
 - of object classes, A-14, A-16

U

- Unicode Transformation Format 8-bit (UTF-8)
- UNIX crypt
 - for password encryption, 16-3, 16-4, 23-7, A-24
 - for password hashing, 16-3
- UNIX, starting Oracle Directory Manager on, 4-2
- unspecified access, 14-10, 14-23
- user
 - login, A-2
 - names and passwords, managing

- by using ldapmodify, 5-10
- by using Oracle Directory Manager, 5-9
- password modification event, 10-9
- user certificates, searching directory for, G-1
- User field, in Oracle Directory Manager, A-2
- User Management Application Administrators Group, 17-10
- User Preferences
 - button, 4-5
 - menu item, 4-4
- User Proxy Privilege Group, 17-13
- usercertificate attribute, G-1
- userPassword attribute, hash values, 23-7
- users
 - entries
 - adding by using ldapadd, 6-8
 - adding by using Oracle Directory Manager, 6-4
 - modifying by using ldapmodify, 6-8
 - modifying by using Oracle Directory Manager, 6-5
 - guest, 5-9
 - names and containment, planning, 19-3
 - proxy, 5-9, 12-4
 - super, 5-9
- UTF-8. See Unicode Transformation Format 8-bit
- UTLBSTAT.SQL, 21-2
- UTLESTAT.SQL, 21-2

V

- Verifier Services Group, 17-12
- View menu, in Oracle Directory Manager, 4-4
- virtual memory, 20-9
- vmstat utility, 21-2

W

- wallets
 - passwords, A-3
- wildcards, in setting access control policies, 14-36
- Windows
 - starting Oracle Directory Manager on, 4-2
 - Task Manager, 21-2
- Windows Performance Monitor, 21-2
- worker threads, 21-7
 - specifying in replication, A-11